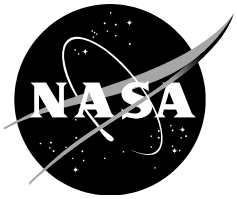
NASA/TP—2012–220450



Methods for Detection and Correction of Sudden Pixel Sensitivity Drops

Jeffery Kolodziejczak
MSFC NASA

Robert Morris
ARC NASA

January 2012

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

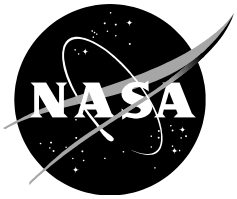
- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TP—2012–220450



Methods for Detection and Correction of Sudden Pixel Sensitivity Drops

Jeffery Kolodziejczak
MSFC NASA

Robert Morris
ARC NASA

National Aeronautics and
Space Administration

January 2012

KPO@AMES DESIGN NOTE

Design Note No:	KADN-26304		
Title:	Methods for Detection and Correction of Sudden Pixel Sensitivity Drops		
Author 1:	J. Kolodziejczak	Signature:	
Author 2:	R. Morris	Signature:	
GS SE Approval:	R. Nevitt	Signature:	
Science Approval:	M. Haas	Signature:	
Distribution:	J. Jenkins, D. Caldwell, S. Bryson		

Revision History:

Rev. Letter	Revision Description	Date	Author/Initials

Design Note No:	KADN-26304	Rev:	Draft	Date:	31-Jan-2012
Title:	Methods for Detection and Correction of Sudden Pixel Sensitivity Drops				
Author:	J. Kolodziejczak & R. Morris				

Overview. This document describes methods used for detection and correction of sudden pixel sensitivity drops (SPSDs) in the PDC CSCI. SPSDs appear as step drops in the signals from pixels caused by damage from energetic particle interaction on the CCD. They are often accompanied by a recovery period, but may exhibit a persistent step. They affect 5-10% of light curves per quarter.

Recommendations. None.

Applicable Documents. None.

Open Items/Action Required. None.

TBDs/TBRs. None.

Contents

1. Introduction

PDC 8.0 includes implementation of a new algorithm to detect and correct step discontinuities appearing in roughly one of every twenty stellar light curves during a given quarter. An example of such a discontinuity in an actual light curve is shown in fig. 1. The majority of such discontinuities are believed to result from high-energy particles (either cosmic or solar in origin) striking the photometer and causing permanent local changes (typically -0.5% in summed apertures) in quantum efficiency, though a partial exponential recovery is often observed [1]. Since these features, dubbed sudden pixel sensitivity dropouts (SPSDs), are uncorrelated across targets they cannot be properly accounted for by the current detrending algorithm. PDC de-trending is based on the assumption that features in flux time series are due either to intrinsic stellar phenomena or to systematic errors and that systematics will exhibit measurable correlations across targets. SPSD events violate these assumptions and their successful removal not only rectifies the flux values of affected targets, but demonstrably improves the overall performance of PDC de-trending [3] .

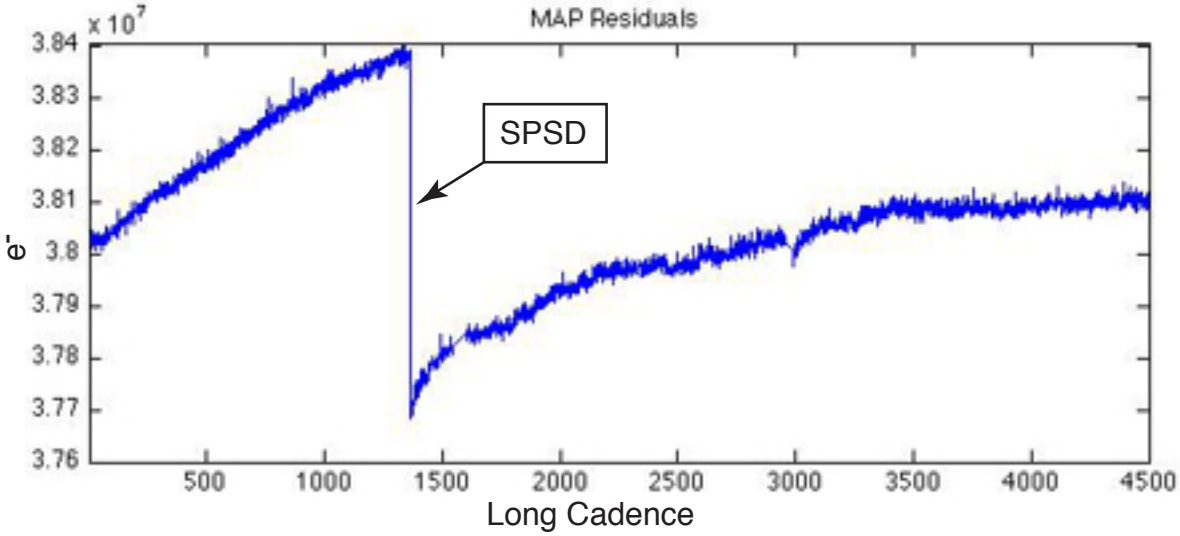


FIGURE 1. **Example sudden pixel sensitivity dropout (SPSD).** The observed $\sim 2\%$ drop in signal is essentially instantaneous, occurring in one of the 270 co-added frames of a long cadence. The drop is often initiated by a detected cosmic ray (CR), which deposits most of its secondary charge in or near the pixel exhibiting the step drop. This suggests that SPSDs result from CR damage, but in some cases the CR-induced charge is deposited in a pixel adjacent to the aperture while the damage is to one or more pixels in the aperture. The time series also exhibits a slow increase in relative signal during the first ~ 100 long cadences following the step drop, which we define as the *recovery* period. This is Q5 data for KID 8414961.

1.1. **Implementation.**

Context. As part of the SOC pipeline, depicted in fig. 2, PDC receives raw light curves and centroids from PA, then corrects systematic and other errors, removes excess flux due to aperture crowding, and conditions light curves for the transiting planet search (TPS). The execution of PDC is managed by the module `pdc_matlab_controller`, which maintains the ability to run the pre-MAP version of PDC. The algorithm described herein, whose execution is managed by the module `spsd_controller`, is only

called from the PDC-MAP module `presearch_data_conditioning_map`, which maintains the ability to alternatively call the pre-8.0 discontinuity module. See fig. 3.

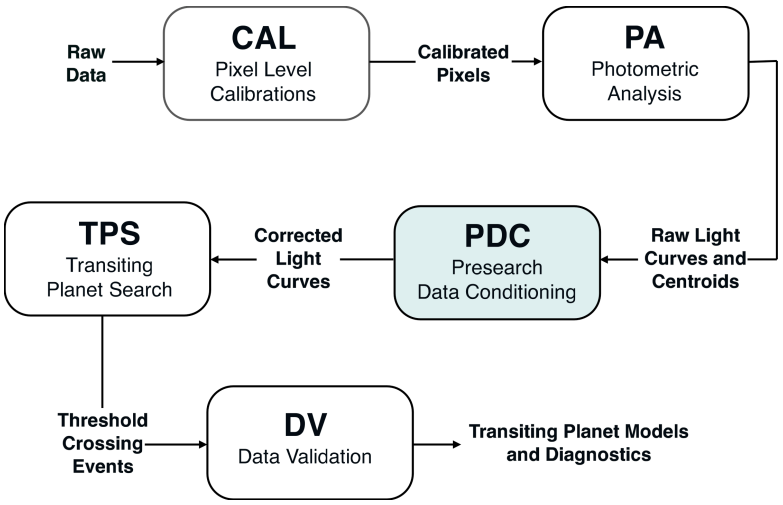


FIGURE 2. **Pipeline data flow.** These SPSD detection and correction algorithms are implemented in PDC. PDC receives raw light curves and centroids from PA, then removes systematic errors to produce corrected light curves for input into TPS.

2. ALGORITHMS

The algorithm consists of three components: filtering, detection, and correction. The first component uses a Savitzky-Golay [4] inspired detection filter which accounts for the step and higher order discontinuities and implements a multi-scale approach to improve localization of the peak response. Detection and correction are performed on all targets in a given channel in an iterative process whereby the maximal detector response for each target is evaluated and, if an SPSD is detected, a correction is applied. For the current iteration, targets in which SPSDs were found and corrected are retained in a list and the process is repeated until the list is empty.

Fig. 4 contains a flow diagram charting the process of deriving the detection filter, detecting the SPSDs and deriving time series corrections based on

- a set of input parameters,
- typically full-quarter flux time series for all targets, which have undergone preliminary systematic-error-corrected using *coarse MAP*, and
- the co-trending vectors used in coarse MAP, $\widehat{U}$.

In this and later flow diagrams the shaded boxes represent items which are described in more detail elsewhere. The part of the algorithm represented by the box labeled "Derive detection filter" is detailed in §2.1, while the "Detect SPSDs" and "Correct SPSDs" boxes are described in §§2.2 and 2.3. The legend at the bottom of the figure applies to all subsequent flow diagrams in this document. Since the algorithm has not yet been adapted to short cadences, we imply "long cadence" when using "cadence."

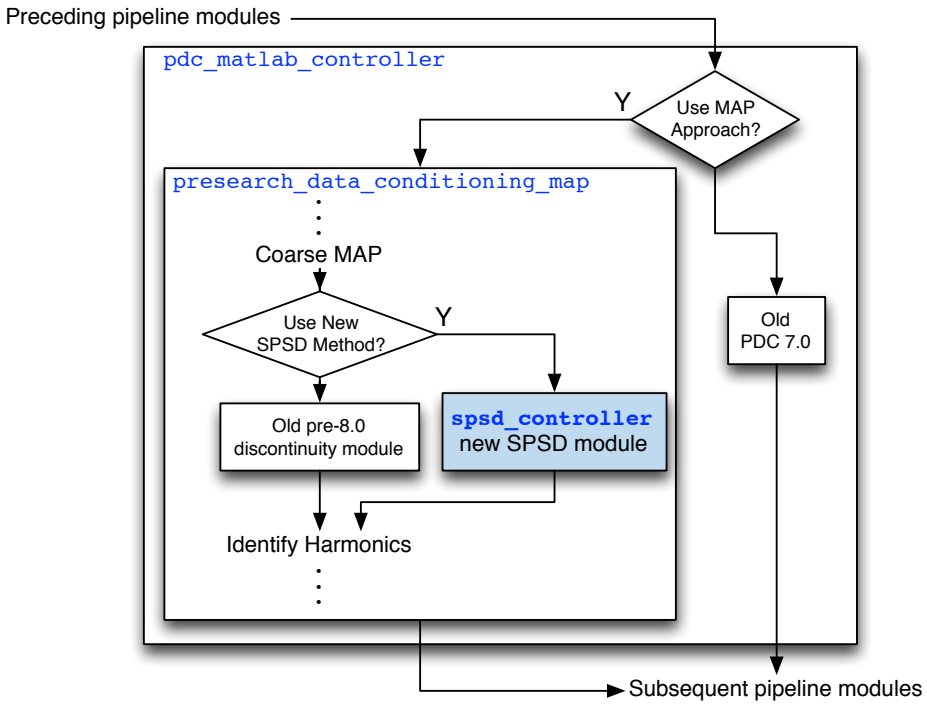


FIGURE 3. **Context of new SPSD algorithm within PDC.** These algorithms are available only when the MAP approach is chosen for PDC, and then only if selected as an alternative to the discontinuity module employed prior to SOC pipeline version 8.0.

2.1. **Detection Filter.**

Overview. We apply a filter to the data to enable the detection process. The light curves exhibit a wide variety of features in all frequency bands due to stellar behavior, transits, and various systematic errors. Therefore, we first need to identify what distinguishes an SPSD from other features. We assume that the signature we are looking for in the SPSDs really has only four distinguishing characteristics:

- it is instantaneous, i.e, a discontinuity,
- it is negative going,
- it is not accompanied by a similar positive-going close-proximity feature indicative of a transit or flare,
- it is relatively uncommon and uncorrelated among targets, occurring in a small percentage of targets at most only a few times per quarter.
- it is common for the discontinuity to to exist, not just in the values (0th polynomial order), but in the slope (1st polynomial order) and quadratic behaviors (2nd polynomial order) of the light curve as well. In fact, the exponential recovery that often follows SPSDs suggests that all polynomial orders are affected.

A conceptual starting point for identifying an SPSD is to imagine fitting a full N-cadence time series to a set of basis vectors which includes a unit step at a specified time, t_i, and a generic orthogonal set, which is

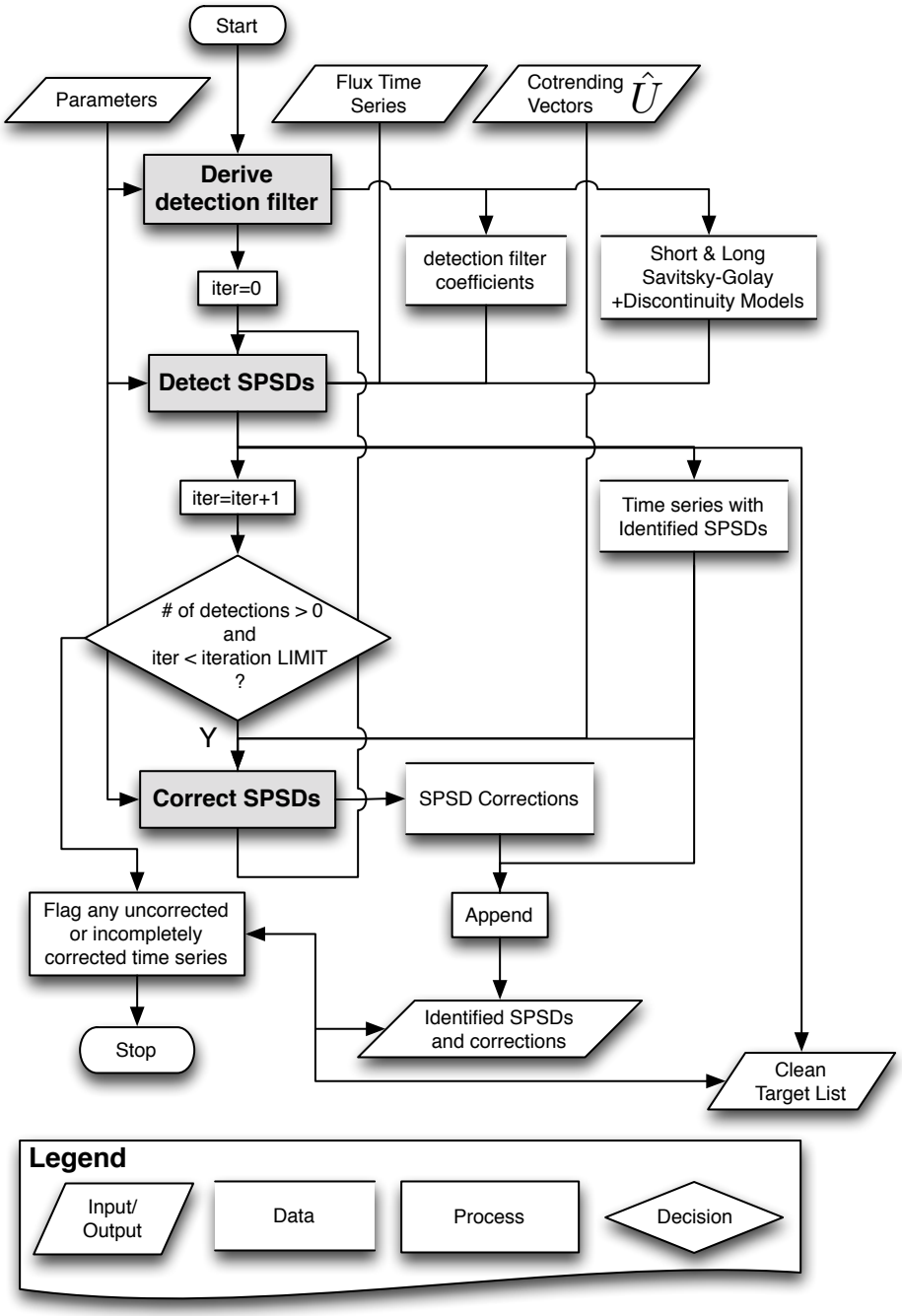


FIGURE 4. **Top-level SPSD process flow.** The algorithm requires a set of parameters, flux time series, and a set of cotrending vectors (possibly $\emptyset$). As its principal results, it produces a set of identified SPSDs, time series corrections, and a list of clean targets. Detection and correction occur iteratively, with at most one SPSD identified and corrected in any time series per iteration. Only time series with identified SPSDs proceed to the next iteration in search of multiple-SPSD time series. Processes in shaded boxes are detailed in following sections, along with associated parameters and intermediate results.

not quite complete, but spans the non-noise frequency components that would still be present in the light curve if the SPSD was absent. If we fit the light curve $N-2$ times, each using a different step component generated by allowing i to increment over $\{2, \ldots N-1\}$, then the coefficients of the step component from each fit combine to represent a time series of $N-2$ step height estimates. The one occurring at the SPSD location is the best estimate of the step height, and has the largest value assuming the step height is significantly larger than the noise. Typical values away from the SPSD represent $0+$noise unless the light curve includes other step-like features, such as transits or flares. This would be an effective but inefficient process for detecting SPSDs. The inefficiency results from a lack of information about the SPSD in data far from the step and the fact that more orthogonal basis vectors are required to fit variations in a given frequency band in a long time series than a shorter one.

Alternatively, we could apply a minimal filter by calculating the first or second differences of the time series to estimate the first or second derivative at the shortest possible time scale. The SPSDs would again stand out, as long as the step height is significantly larger than the noise. This would be a much more efficient but less effective method since the only the information immediately surrounding each time is used, so the noise would be larger. The relative noise increase depends on the frequency content of the light curve.

In the algorithm described below, we have strived to strike a balance between the effectiveness of modeling and the efficiency of filtering. We use a model to derive a set of filter coefficients which give a step height estimate when applied to the data. The model is linear, and the component of the left inverse or pseudo-inverse of the design matrix which corresponds to the unit step component of the model provides the coefficients. Unlike the model described above, the length of the filter is limited to a range which contains information about the SPSD step, and the filter only measures the step height rather than fitting the entire set of model coefficients. Since the filter takes other model components into account, it include the information from a wider range of data in the step height estimate and therefore provide better signal-to-noise than the minimal filter.

Any filter has a step response function, which introduces a transient response into the filtered time series around the location of the step. In addition, there are a variety of relevant time scales and models which may be more effective for one light curve than another and even from one time to another within a single light curve. To address both of these issues we have introduced a multi-scale approach for calculating the filter parameters. Starting with a given time-scale and model, the algorithm iteratively calculates a series shorter time scales and simpler models and weights, whose weighted sum of associated filter coefficients reduces the transient step response as much as possible, while maintaining the signal-to noise at the original scale. The result of applying the weighted average of these multi-scale coefficients is equivalent to fitting the time series locally to all the model-scale combinations for the step heights and then taking the weighted average of the step heights, only much more efficient to calculate. The resulting filter has a step response function which is much closer to the ideal (a δ function) than that derived from any single model, and it maintains its effectiveness in the vast majority of light curves. The output of the filter is still simply a time series of modeled step heights.

Parameters. The input filter parameters define 3 bounding models used in the generation of the filter coefficients. The values are either a length, a polynomial order, or a discontinuous polynomial order. The models are described below. Table 1 lists them and cross-references with variable names in the code.

Parameter	Description	Default	MATLAB Name
$N_{\mathrm{LongWindow}}$	length of long-window(LW) model	193	`.windowWidth`
n_{LWpoly}	polynomial order of LW model	3	`.sgPolyOrder`
$n_{\mathrm{LWdiscontinuity}}$	polynomial order of discontinuity in LW model	2	`.sgStepPolyOrder`
$N_{\mathrm{ShortWindow}}$	length of short-window(SW) model	11	`.shortWindowWidth`
n_{SWpoly}	polynomial order of SW model	1	`.shortSgPolyOrder`
$n_{\mathrm{SWdiscontinuity}}$	polynomial order of discontinuity in SW model	1	`.shortSgStepPolyOrder`
N_{min}	length of minimal model	9	`.minWindowWidth`

TABLE 1. **Detection Filter Parameters.** These parameter specify three bounding models, long-window, short-window and minimal, in the algorithm described below. The minimal model polynomial order and polynomial order of discontinuity are both set to 1. The dot in front of the MATLAB name is to indicate the variables are a contained in structures.

Input/Output. The algorithm has no inputs, its principle output is a vector of filter coefficients, K. It also outputs the design matrix, generically M, and left inverse, A, of both the short- and long-window models for use in the validation process. See Table 2.

Output Variable	Description	MATLAB Name
K	$N_{\mathrm{LongWindow}}$-vector of filter coefficients	`.kernels`
M^L	long-window model design matrix	`.longModel.designMatrix`
A^L	left inverse of M^L	`.longModel.pseudoinverse`
M^S	short-window model design matrix	`.shortModel.designMatrix`
A^S	left inverse of M^S	`.shortModel.pseudoinverse`

TABLE 2. **Detection Filter Outputs.** The algorithm outputs the multiscale filter coefficients, K as a vector of length, $N_{\mathrm{LongWindow}}$ cadences. If n is the number of linear terms, and therefore, coefficients in a model, and n is the length scale of the model, e.g. $N_{\mathrm{LongWindow}}$ or $N_{\mathrm{ShortWindow}}$, then M is $m \times n$ and A is $n \times m$.

Description. The design of the linear shift-invariant detection filter is based on the method of Savitzky and Golay [4]. The algorithm's flow diagram is shown in fig. 5. We model each N_{win}-cadence window (N_{win}, odd) of the light curve as $y = Mc$, where M is a matrix formed by assembling the n N_{win}-element basis vectors of the model into columns of an N_{win} row $\times$ n column model component matrix or *design matrix*, and c is an n-element column vector. For the purposes of this document we always let the first column of M represent an antisymmetric unit up-going step function with zero center value, to model the SPSD's step discontinuity. Under this construction, we compute the *left inverse* of M, which is a pseudoinverse obtained by multiplying from the left, $(M^TM)^{-1}M^TMc = c = (M^TM)^{-1}M^Ty$, so letting $A = (M^TM)^{-1}M^T$, means $c = Ay$. A is a $n \times N_{\mathrm{win}}$ matrix and we identify the detection filter coefficients as the first row of A, $A_{1,j}$, because for a given N_{win} cadence range of the light curve indexed by j, y_j, the best fit step height at the window center is $\sum_j A_{1,j}y_j$. Model specifics are discussed in the paragraph labeled "**Model**" below.

We generate filter coefficients at multiple scales, where *scale* is simply the length of the filter. The scales are selected to reduce the step response of the combined filter as it scans across a step while still

measuring the step height when centered at the location of the step. When combining a shorter scale with a longer scale, we symmetrically pad the ends of the shorter coefficient vector with zeroes, keeping the centers aligned, and we weight the coefficients in the shorter scale model by a relative factor w to prevent an increase in the ratio of filter response to white noise far from a step, to the filter response to a unit step at the step. This multi-scale segment of the process, defined by the shaded boxes encompassing circular markers A-B-C in figs. 5 and 7, is discussed further in the paragraph labeled "**Multiscale Segment**" below.

Model. At each point t_i in time, we model the data within an $N_{\rm win}$-length window centered at t_i as a linear combination of basis functions f_0, f_1, ..., f_n. At each point we find the coefficients a_0, a_1, ..., a_n that yield the best fit in the least squares sense. One of the basis functions, f_s, is a unit step and its corresponding coefficient, a_s, is a measure of the step function's contribution to the best-fit mixture. The value of this coefficient is our basic detection statistic.

The basis for the model we use to derive the filter coefficients consists of vectors designed to describe three main elements of a $N_{\rm win}$-cadence-length window of the light curve centered at time t_i. These elements are

a step discontinuity at t_i: This is simply the discrete, antisymmetry step function, Θ_j, where

$$\Theta_j = \left\{ \begin{array}{rcl} -0.5 & \forall & i - \lfloor N_{\rm win}/2 \rfloor \le j < i \\ 0 & \leftrightarrow & j = i \\ 0.5 & \forall & i < j \le i + \lfloor N_{\rm win}/2 \rfloor \end{array} \right\}.$$

We specify that all other basis vectors in the model be continuous at zero-order so that the coefficient corresponding to this step is the least-squares fit estimate of the instantaneous SPSD step height. The filter output is a weighted average of step height coefficients from a series of models.

recovery-related higher order discontinuities at t_i: If an SPSD occurs at t_i, then in most cases an exponential transient recovery period begins at t_i, resulting in not just a step, but a discontinuity at all Taylor series orders of the light curve at t_i. Since the Taylor series expansion terms are falling as $1/(\Delta t^n n!)$ with order, n and time constant $\Delta t \gg 10$, it is generally sufficient to model only the first few terms, e.g. a slope and quadradic discontinuity, to produce a good measure of the instantaneous step height. For the general case of $n_{\rm discontinuity}$ orders, we model the the recover with $n_{\rm discontinuity}$ terms, Φ_j^n, where n is the order and j is the time index, as follows

$$\Phi_j^n = \left\{ \begin{array}{rcl} 0 & \forall & i - \lfloor N_{\rm win}/2 \rfloor \le j \le i \\ P_n\left(\frac{2(j-i)}{N_{\rm win}-1}\right) - P_n(0) & \forall & i+1 < j \le i + \lfloor N_{\rm win}/2 \rfloor \end{array} \right\},$$

with P_n, the Legendre polynomials of order n. The range is scaled to span $[-1, 1]$ across the full window width and the offset $-P_n(0)$ to ensure zero order continuity at t_i. Without these terms, the step height would generally be underestimated and the SPSD signal-to-noise of the filtered data would be worse. The continuity at t_i minimizes the bias these terms can contribution to the zero-order step.

the unaffected continuous component: When no SPSD is present we model the segment of the light curve as a smooth continuous function composed of $n_{\rm poly}$ polynomial terms, Ψ_j^n, defined by:

$$\begin{array}{rcl} \Psi_j^0 & = & 1 \\ \Psi_j^n & = & P_n\left(\frac{2(j-i)}{N_{\rm win}-1}\right) - P_n(0) \end{array} \quad \forall\ i - \lfloor N_{\rm win}/2 \rfloor \le j \le i + \lfloor N_{\rm win}/2 \rfloor.$$

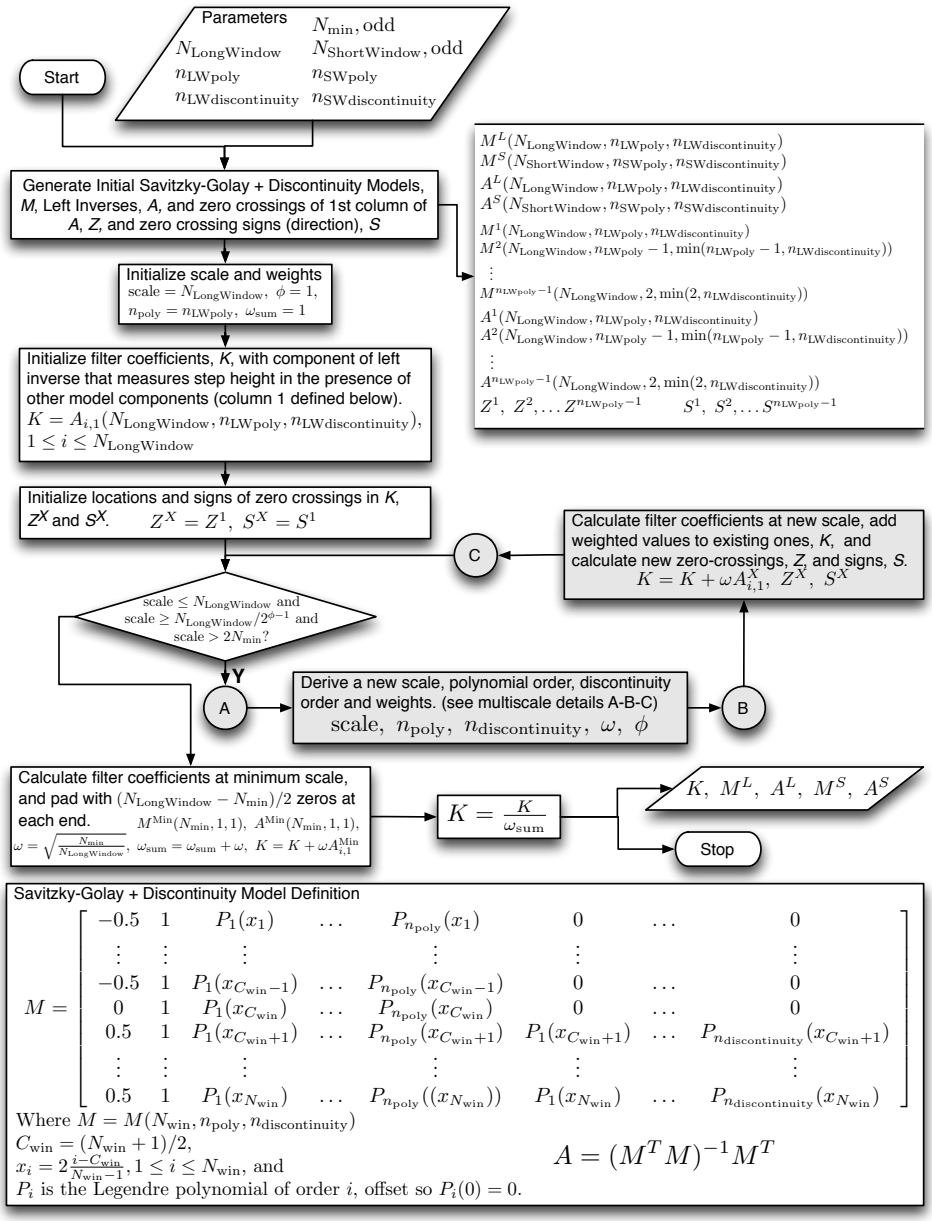


FIGURE 5. **Detection filter derivation algorithm.** The algorithm requires a set of parameters describing 3 Savitzky-Golay + discontinuity models from which the filter is derived. Each model requires a length in cadences, $N_{\rm win}$, a polynomial order for the S-G components, $n_{\rm poly}$, and a polynomial order for the discontinuity, $n_{\rm discontinuity}$. The parameter sets $\{N_{\rm LongWindow},\ n_{\rm LWpoly},\ n_{\rm LWdiscontinuity}\}$ and $\{N_{\rm ShortWindow},\ n_{\rm SWpoly},\ n_{\rm SWdiscontinuity}\}$ specify long and short window models, respectively, and $N_{\rm min}$ is the length of a minimal model with $n_{\rm poly} = n_{\rm discontinuity} = 1$. The algorithm produces a multiscale filter kernel, K, $N_{\rm LongWindow}$ cadences long, along with model component matrices M^L and M^S and left inverses (pseudo-inverses) A^L and A^S for the long and short cases. Detection filter construction occurs recursively using a multiscale approach designed to maintaining good SPSD signal-to noise with highly localized response in a wide variety of light curves. Processes in shaded boxes (A-B-C) are detailed in following sections. See figs. 4 and 7.

Here again, we choose to offset by $-P_n(0)$, which leads to interpretation of the constant term, which is the only non-zero term at the window center, as the least-squares fit estimate of the light curve window's central value. This has no effect on the step height, but we use the constant coefficient of this fit to estimate noise in the algorithm's validation segment.

As indicated in more detail in the box at the bottom of fig. 5, These basis elements are combined to form a design matrix, $M = \{\Theta, \Psi, \Phi\}$. We then proceed to calculate A, and the desired filter coefficients $A_{1,j}$ as discussed above and illustrated graphically in fig 6. If the polynomial orders and window length are appropriately chosen, this model produces a good fit for nearly all light curve and times. Far from an SPSD the step height coefficient is estimating zero, and as a result is measuring the noise-induced error in the fit step height. At the exact time of the SPSD, the step height coefficient is an estimate of the instantaneous step, with any other potentially biasing features accounted for by the other model terms. When an SPSD is in the window but not at the center the model is incorrect. There is some mixing between other terms in the model and the unit step, which results in a smaller absolute step height. The maximum absolute step height will, therefore, always occur at the true time that the SPSD occurred, but there will be a signal from the filter in the range equal to the filter window width around the step which corresponds to the filter's *step response function*.

To account for the fact that all light curves are not the same, we could generate a bank of filters with different length scales, and orders, and apply them individually to the data, and then combine the results in some sort of weighted average to find the SPSDs. We could have used a shotgun approach for the parameter and weight determination, selecting somewhat arbitrary values for each to sample the parameter space at a variety of time scales, or a wavelet approach which discretely reduces the scales by successive factors of two. However, to maintain processing efficiency we have developed the multiscale approach discussed in the following paragraphs. This method selects and combines various scales into a single filter with reduced step response when the SPSD is in the window but not at the center, while maintaining the centered SPSD signal-to-noise ratio (SNR) at the level attained at long time scales. In other words the method drives the step response function toward being a *delta*-function without completely ignoring data more remote from the step.

In addition to multiscale coefficients, the algorithm generates the full model matrices and left inverse for both the long and short cases, which are used in the SPSD validation process. For these we add three additional basis vectors which are simply discrete (Kronecker) δ-functions located at the center and ± 1 cadence from the center. Their purpose is to exclude the central three data points from affecting the other fit coefficients when applied during the validation phase. (Equivalently, we could have excluded the central three points from the fit.)

Multiscale Segment. The final SPSD detection filter is produced by constructing filters using the method described above at multiple scales and adding them together to improve localization of the peak response. The segment A-B-C in fig. 5 determines the model parameters and weights for the succession of scales. The details of the process are include in fig. 7. During initialization we derive a set of filter coefficient vectors at the longest time scale, $N_{\mathrm{LongWindow}}$ with various polynomial and discontinuity orders.

We observe that the filter response for a given set of filter coefficients has peaks and valleys with the peaks corresponding to places where the filter coefficents cross zero from minus to plus with time and the valleys corresponding to where they cross from plus to minus. A way to drive the the step response function toward a *delta*-function is to choose parameters for the new filter so that the outermost valley of the new step response function lines up with the outmost peak of the initial one. The information we need to make this choice comes from a catalog of the filter coefficient zero crossing locations, Z_k^i and

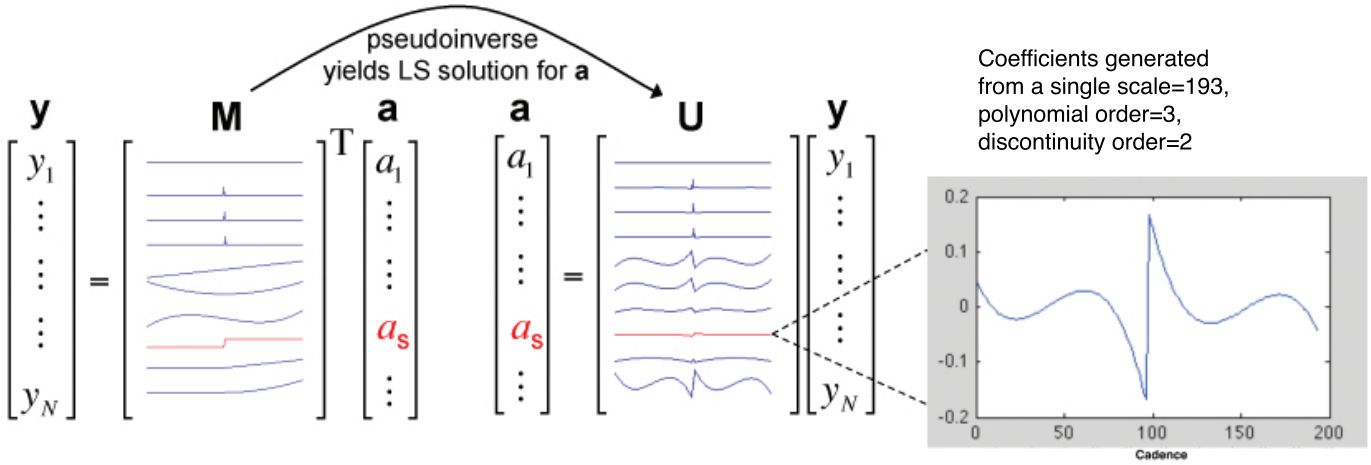


FIGURE 6. **Conversion of model basis vectors for a time series window to filter coefficients.** A time series segment y is modeled by basis vectors assembled as columns in a matrix, M, and a coefficient vector a. The pseudoinverse of M, U provides the least-squares solution for a when multiplied by y. The coefficient, a_s, of the unit step basis vector is the step height, and the row of U which give a_s when multiplied by y contains our filter coefficients.

signs (minus-to-plus= $+1$, or plus-to-minus= -1) of zero crossings, S_k^i, where i is the index to a filter coefficient vector, and k is the index to a discrete zero crossing within a filter coefficient vector. The index i corresponds to specific values for the pair of parameters $\{n_{\rm poly}, n_{\rm discontinuity}\}$ as shown in the upper right in fig. 5. We always maintain $n_{\rm discontinuity} \leq n_{\rm poly}$ to reduce the size of the catalog. Since the model always produces filter coefficients which are antisymmetric with respect to the center, only the zero-crossings on one side of the center need to be catalogued.

For each multiscale loop iteration, we have a current state X which includes the addition of a set of filter coefficients $A_{1,j}^X$ from the previous scale and calculated zero-crossing locations, Z_j^X, and zero-crossing signs, S_k^X, of the *combined* filter coefficient vector at that state. To determine what scales of a given model, i_1, would improve the step response function, we assemble a ratio matrix, R, where

$$\mathrm{R}_{jk} = 2\left\lfloor \frac{N_{\rm LongWindow} Z_j^{\rm X} S_j^{\rm X}}{2 Z_k^{i_1} S_k^{i_1}} \right\rfloor + 1,$$

representing all the integer scales where zero-crossings of the candidate model align with those of the current state. We chose from R a subset, R', whose elements are negative, meaning zero crossings are in opposite directions, and larger in absolute value than $N_{\rm LongWindow}/2^{\phi}$, where ϕ is the multiscale iteration number starting at 1. This admits shorter scales with increasing iteration. When possible we consider 2 candidate models one with the current value of $n_{\rm poly}$ and one with decremented polynomial order, unless the current state is already the minimum to be considered, $n_{\rm poly} = 2$. We select the least negative (maximum) value from these candidates as the new scale and if it comes from the lower order set we reduce $n_{\rm poly}$ by one, and if necessary, reduce $n_{\rm discontinuity}$ as well. The parameters $\{scale, n_{\rm poly}, n_{\rm discontinuity}\}$ specify a new model from which a new set of filter coefficients are derived. The process iterates as long as a scale can be found in the range $[N_{\rm LongWindow}/2^{\phi}, N_{\rm LongWindow}]$, and finally stops once scale falls below $2N_{\rm min}$. It then adds a final step at scale= $N_{\rm min}$ with 1st order polynomials.

The noise level of the filter output is approximately proportional to $1/\sqrt{\text{scale}}$. It is therefore possible to maintain the SNR of the multiscale filter by weighting coefficients at each scale by a factor $\omega/\omega_{\text{sum}}$, where $\omega = \sqrt{\text{scale}/N_{\text{LongWindow}}}$ and ω_{sum} is the sum of ω over all scales used. The result therefore represents a weighted average of least-squares fit step heights yielding a combined multiscale step height estimate. The construction of the filter using default parameter values is illustrated in fig. 8, and the final filter is plotted in fig. 9. Refer to §3.1 for discussion of performance characteristics of this filter.

2.2. **Detection Algorithm.**

Overview. The detection process begins with preconditioning of each time series by filling gaps and padding endpoints. Conditioned time series are then convolved with the detection filter and responses are normalized across all targets and cadences on the current channel. The maximum normalized detector response for each target is evaluated by applying succession of threshold-based decisions to determine first whether it is a feasible candidate and finally whether it should be labelled an SPSD. Fig. 10 is a flow diagram of the detection algorithm described below. The shaded boxes are further diagramed in figs. 12 13, and 14, with some details relegated to Appendices.

Parameters. The input detection parameters are either thresholds or used to calculate thresholds. The values are either ratios or probabilities. The threshold calculations are described in Appendix B. Table 3 lists the parameters and cross-references with variable names in the code.

Parameter	**Description**	**Default**	`MATLAB` **Name**
f_{FP}	The specified rate of false positives due to noise	0.005	`.falsePositiveRateLimit`
R_{steps}	Tolerance for ratio of long- to short-model step height estimates	0.7	`.discontinuityRatioTolerance`
$R_{\text{min:max}}$	Tolerance for $\lvert \log \text{min/max} \rvert$ within a window for transit veto	0.7	`.transitSpsdMinmaxDiscriminator`

TABLE 3. **Detection Parameters.** These parameter specify constraints on SPSD acceptance. The false positive rate is used to calculate detection thresholds for identifying candidates (see Appendix B). The other two parameters are used for validation of SPSDs as discussed below.

Input/Output. The inputs for the algorithm include the filter coefficients and models listed in Table 2 as outputs of the detection filter generation algorithm and a set of time series, listed in the short Table 4. The first time the algorithm is called, time series for all targets output from coarse map are used. On subsequent calls only time series in which SPSDs have previously been found and corrected are input. The algorithm's principle output is a list containing the subset of input targets in which SPSDs were identified, and its complement the list of "clean" targets, as well as some parameters which characterize the SPSDs. See Table 5.

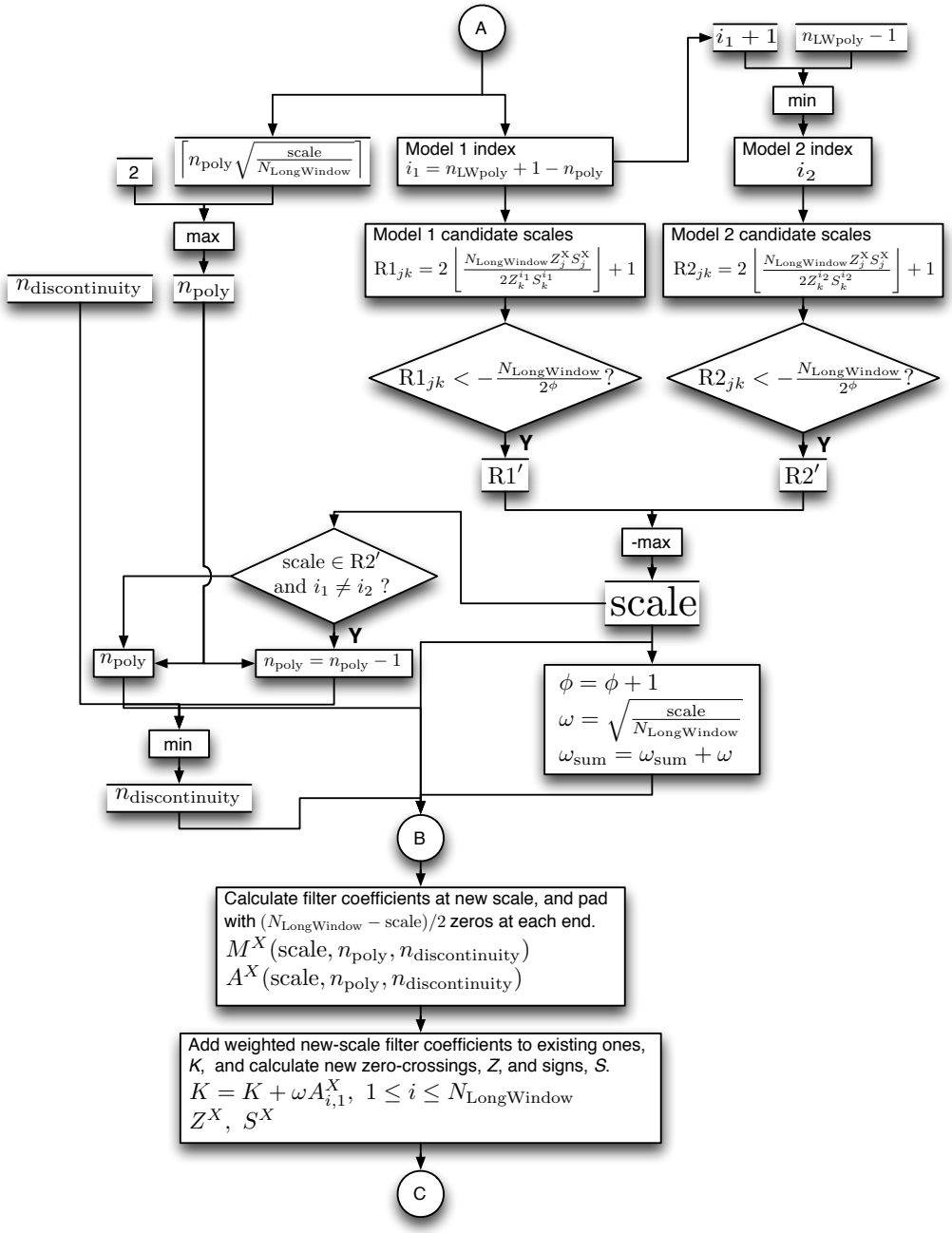


FIGURE 7. **Multiscale segment (A-B-C) of detection filter derivation algorithm.** This algorithm segment determines a new length in cadences, scale, a polynomial order for the S-G components, n_{poly}, and a polynomial order for the discontinuity, $n_{\mathrm{discontinuity}}$ with which to build the next in a series of weighted summed filter components. It then generates the model component matrices M^X and and left inverse A^X and A^S for specified cases. It selects the scale by matching the zero crossing points at the current state of the summed detection filter with the opposite-going zero-crossings of a pair of candidate models, one of current polynomial order and one of reduced order. These zero-crossing ratios are tabulated in the matrices $\mathrm{R1}_{jk}$ and $\mathrm{R2}_{jk}$ and the scale selected is the largest one identified in a range which is reduced from the full length, $N_{\mathrm{LongWindow}}$, by a factor of 2 on each iteration, ϕ. See fig. 5.

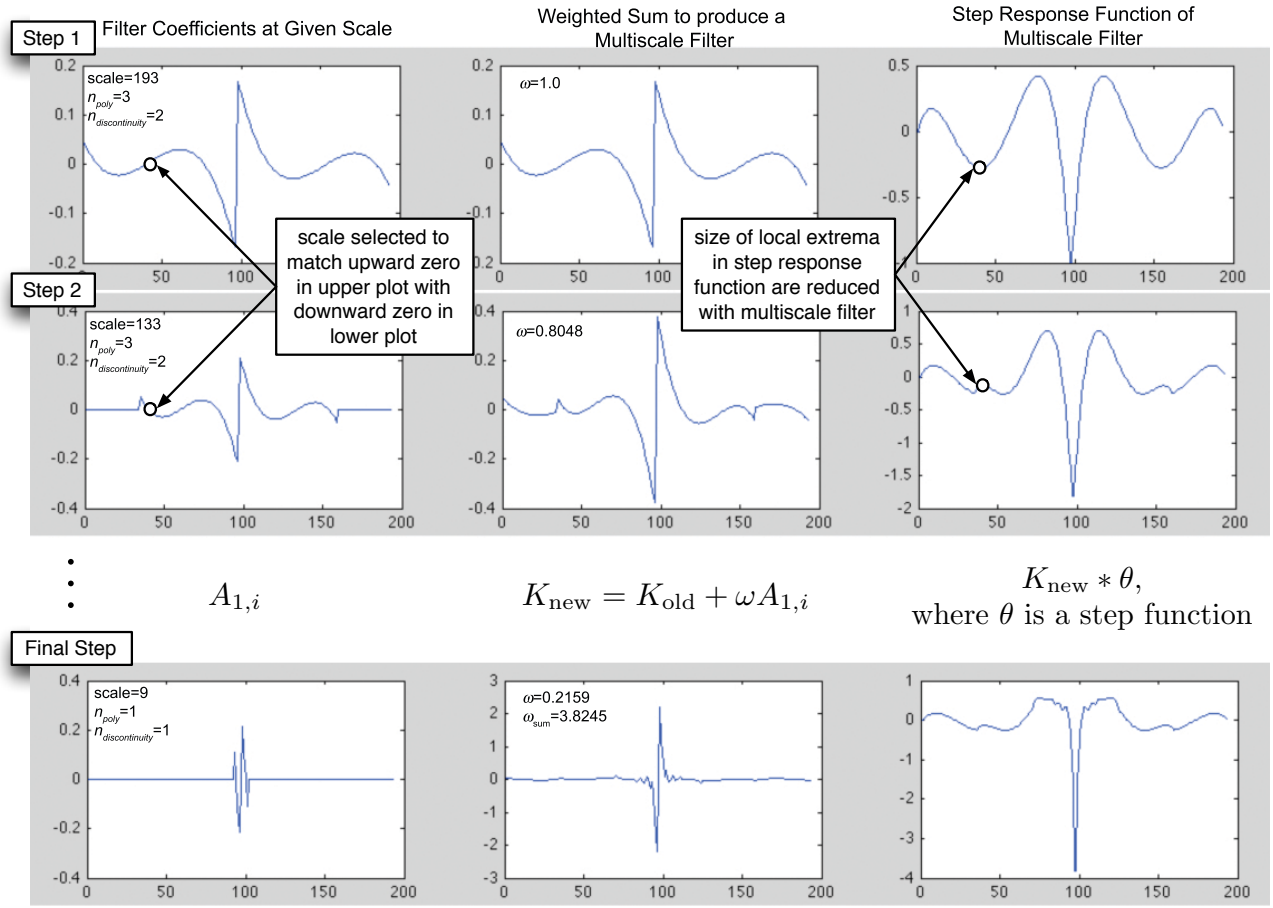


FIGURE 8. **Illustration of multiscale filter construction.**

Input Variable	Description	MATLAB Name
ts	set of flux time series	`.fluxResiduals`
g	gap indicators for time series	`.gaps`

TABLE 4. **Detector Inputs.** The algorithm requires a set of flux time series and associated gap indicators.

Description. The main steps in the algorithm shown in fig. 10 occur serially and only once per call. After an initial check to ensure that input time series contain valid data, we propagate the {OK} time series through a chain of operations:

$$ts \xrightarrow{\text{precondition}} ts^{P} \xrightarrow{\text{filter}} ts^{D0} \xrightarrow{\text{standardize by target}} ts^{D1} \xrightarrow{\text{standardize by cadence}} ts^{D2} \xrightarrow{\text{restandardize by target}} ts^{D3}.$$

The preconditioning operation, segment D-E, is described below. Once preconditioned, ts^P are padded by $\lfloor N_{\text{LongWindow}}/2 \rfloor$ cadences and de-gapped in a manner which enables effective filtering, and therefore effective sensitivity to SPSDs near the start, end and gaps. The filter is applied by convolution where $ts^P \otimes K = ts^{D0}(i) = \sum_j u(j)K(i+1-j)$ so a given time series segment is multiplied by the antisymmetric filter coefficient vector, K, in the reverse of the order it would be applied in matrix multiplication used in a fit. So, the sign of the step height is reversed and a SPSD's downward step leads to a positive filter

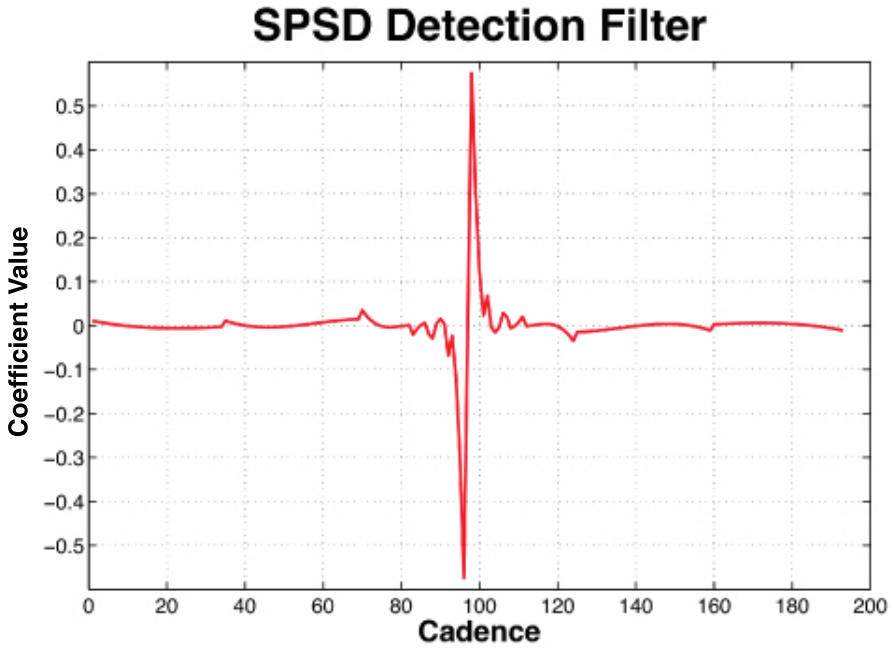


FIGURE 9. **Resulting filter coefficients using default parameter values.**

Output Variable	Description	MATLAB Name
k, {SPSDs}	index to identified SPSD-affected targets	`.spsds.index`
t_k^{ε}	occurrence time of SPSD for target k	`.spsds.spsdCadence`
c_k^L	long-window model (LW) fit coefficients	`.spsds.longCoefs`
h_k^L	LW step height	`.spsds.longStepHeight`
σ_k^L	LW MAD of residuals	`.spsds.longMADs(1,:)`
η_k^L	LW MAD of residual differences	`.spsds.longMADs(2,:)`
c_k^S	short-window model (SW) fit coefficients	`.spsds.shortCoefs`
h_k^S	SW step height	`.spsds.shortStepHeight`
σ_k^S	SW MAD of residuals	`.spsds.shortMADs(1,:)`
η_k^S	SW MAD of residual differences	`.spsds.shortMADs(2,:)`
k, {clean}	index to clean targets	`.clean.index`

TABLE 5. **Detector Outputs.** The algorithm outputs a list of parameters for every confirmed SPSD identifying which target, when and a set of parameters derived from the short-model and long-model fits that are performed on each candidate SPSD. This algorithm finds no more than one SPSD in any input time series per call and its called iteratively to detect multi-SPSD cases.

output value. We keep only those elements of the convolved time series where the filter coefficients vector was applied in full, in other words j in the sum spans the full range of K. As a result the detection time series, ts^{D0}, returns to the original time series length, N. The interpretation of $ts^{D0}(t)$ is the negative of the fit step height at time t, in flux units. Fig. 11 illustrates the filter application on an actual time series.

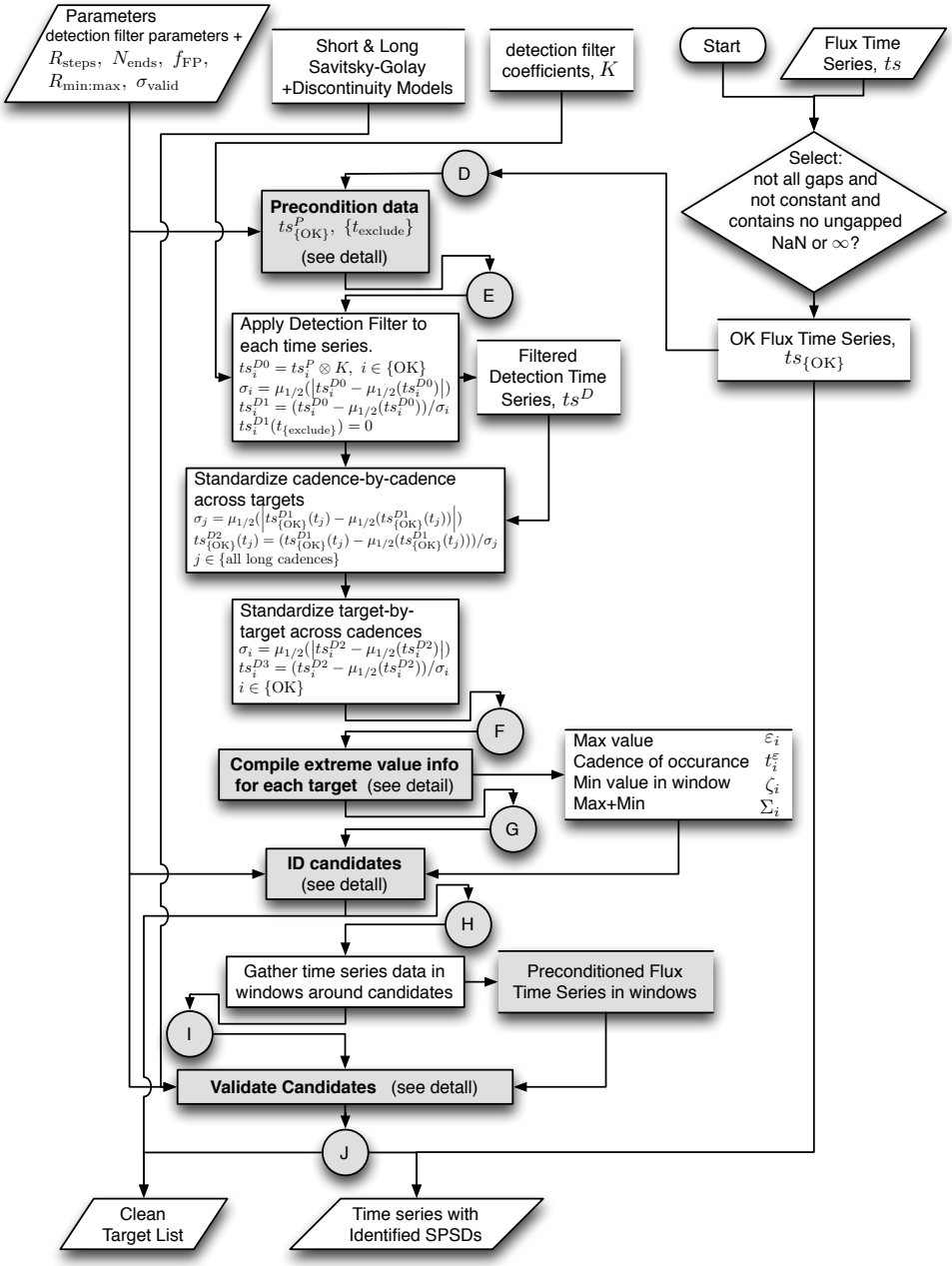


FIGURE 10. **SPSD detection algorithm.** The algorithm requires both the input parameter and the output of the detection filter derivation algorithm shown in fig. 5, as well as an additional set of detection parameters, and a set of flux time series, ts. As implemented, ts have been partially corrected with a coarse PDC-MAP pass [2]. The detection parameters are: a tolerance for the ratio of steps measured using the long:short window models, $R_{\rm steps}$; the window length in cadences for modeling the time series ends used in generating padding data, $N_{\rm ends}$; the accepted fractional rate of false positive SPSD detections, $f_{\rm FP}$; a tolerance for the ratio of maximum detection statistic to nearby minimum detection statistic for transit rejection, $R_{\rm min:max}$; and the threshold for measured step height detection in units of the 1σ shot noise limit $\sigma_{\rm valid}$. It produces a set of identified SPSDs, and a list of clean targets. Processes in shaded boxes (D-E, F-G-H, and I-J) are detailed in following sections. See fig. 4.

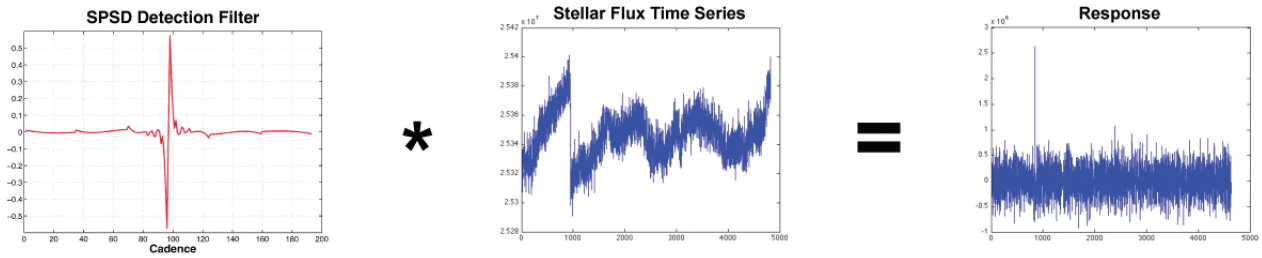


FIGURE 11. **Illustration of filter application.** The filter results in a clear SPSD detection (positive spike), with little evidence of the neighboring step response function.

We want to select SPSD candidates on the basis of the probability that they are real SPSDs and not false positives resulting from statistical fluctuations. To facilitate this, we perform a series of standardization steps on the detection time series. Because Poisson and other sources of noise vary from target to target, we first robustly standardize each detection time series by subtracting the median, $\mu_{1/2}(ts_i^{D0})$. and dividing by a robust estimate of the standard deviation, $\frac{\pi}{2}\times$ the median absolute deviation (MAD), $\mu_{1/2}(\left|ts_i^{D0} - \mu_{1/2}(ts_i^{D0})\right|)$. This also tends to de-weight SPSD detection for targets with a strong component at higher frequencies in the filter bandpass. At the same time we zero the values in a short buffer zone of 5 cadences from each end, and around gaps longer than 1 cadence. These are regions which are prone to residual systematics, which would lead to an unacceptable level of false positives, and by "zeroing them out", we are eliminating the possibility an SPSD will be detected there. This buffering process only amounts to ~ 30 cadences per quarter, affecting the detection efficiency by $< 1\%$. The resulting ts^{D1} all have ~ 0 median and standardized noise levels.

The systematic error removal at this stage of pipeline processing in incomplete and leaves common mode features in the time series. To prevent these features from producing false positives, we perform a cadence-by-cadence standardization step across targets. If the median and standard deviation of the ts^{D1} values across all targets at a given time are significantly different from 0 and 1 respectively, it suggests that systematics are biasing the step height estimates, so we repeat the process described above on the transposed matrix of targets $\times$ cadences to equalize the noise across time, yielding ts^{D2}. The final step, yielding ts^{D2}, is just a repeat of the first target-by-target standardization step, typically resulting in only a small incremental adjustments to median and σ changes induced by the $ts^{D1} \rightarrow ts^{D2}$ step. Note that it is implicit in this algorithm that the input set contains only a small fraction of targets which exhibit variability with a strong high frequency component, otherwise these can bias the standardization parameters and lead to unpredictable results. Also, the second and third steps can only be effective if there is a statistically meaningful sample of targets, so if the number of targets in the set {OK} is ≤ 3 we simply set $ts^{D1} = ts^{D2} = ts^{D3}$.

We use the standardized detection time series, ts^{D3}, to identify candidate SPSDs based on extreme value theory. The details of this algorithm segment, labeled F-G-H in fig. 10, are discussed in the paragraphs below and diagrammed in fig. 13. A part of the algorithm differentiates between SPSDs and transits, enabling the detection of SPSDs which are smaller than transits in the light curve. The result is a set of candidates, $k \in \{\text{SPSD candidates}\}$, who probability of being a false positive due to statistical fluctuations is $< f_{FP}$, along with the maximum values, ε_k, and cadences of occurrence, t_k^{ϵ}.

To further reduce the risk of misidentifying an SPSD because of a rapidly, but not instantaneously changing stellar signal, we include a final validation step. This examines and compares the characteristics of a fit of data in a short window surrounding the candidate SPSD with that of a long window, using the

models defined in §2.1. Candidates that survive these consistency checks are confirmed SPSDs and are output along with the quantities listing in the *Input/Output* section above. Further details are included below and in fig. 14.

Preconditioning. Fig. 12 is a flow diagram of the preconditioning segment (D-E) of the detection algorithm. To simply notation for this discussion we consider the process with input time series for the kth target, and set $F_i = ts_k(t_i)$ and output padded time series $P = ts_k^P$. The purpose of preconditioning is to minimize SPSD detector response near time series continuity breaks. The preconditioning algorithm accepts

- a flux time series vector, F_n,
- a logical gap map vector, g_n,
- and a padding length, $L_{\rm pad} (= \lfloor N_{\rm LongWindow}/2 \rfloor)$,

and produces

- a padded time series vector, P_m,
- and a logical exclusion mask list {exclude},

where

- n runs from 1 to the length of F_n, N,
- m runs from 1 to $N + 2L_{\rm pad}$.

Preconditioning is achieved by dividing elements of P_m into several categories, each of which is generated by a distinct algorithm. These are described in appendix A and summarized below:

Not Gaps: Direct transfer of data for non-gap cadence intervals such that $P_{i_{ng}} = F_{i_{ng}-L_{\rm pad}}$ where i_{ng} are non-gap indices (g_n = false) offset by the length of the pad interval prior to the start of the flux time series.

Length = 1 Gaps: For each gap of length 1 long cadence we proceed as in fig. 33, producing values P_j, where $j \in \{i_{g1}\}$ the list of offset one-cadence gap indices.

Length > 1 Gaps: For each multi-cadence gap we proceed as in fig. 34, producing the set of values $\{P_{\rm gap:start}, , \ldots P_{\rm gap:end}\}$, where gap $\in \{i_{\rm gap}\}$ the list of offset multi-cadence gap indices.

End Padding: We pad the start and end of P as shown in fig. 36, producing the set of values $\{P_1, \ldots, P_{L_{\rm pad}}, P_{N+L_{\rm pad}+1}, \ldots, P_{N+2L_{\rm pad}}\}$.

Outliers: We perform a simple differential 3σ outlier detection and correction as shown in fig. 35, producing P_j, P_{j+1},where $j \in \{\rm out\}$, the list of detected outliers.

The outlier correction is designed to leave steps associated with SPSDs in place, while removing smaller cosmic ray and noise spikes which increase the noise in the filtered output, but are not associated with SPSDs.

The algorithm also produces a list of cadences to exclude when searching for SPSDs. This is formed from the union of the first 5 cadences, last 5 cadences, and windows encompassing a range from 5 cadences before- to 5 cadences after- each multi-cadence gap. As discussed in the *Description* section above, these cadences are later set to zero in the detection time series to prevent false positives due to systematic errors which often appear at or near data gaps.

Extreme Values and Candidate Identification. Fig. 13 is a flow diagram of the extreme value determination and SPSD candidate identification segments (F-G-H) of the detection algorithm. The method is based on the observation that SPSDs are relatively rare events, occurring on average less than once in every 10^5 long cadences, or 5 years, for any one target at typical radiation levels. Because of this, we can just examine the extreme values for the detection statistic, ts^{D3}, for any target and quarter, and evaluate

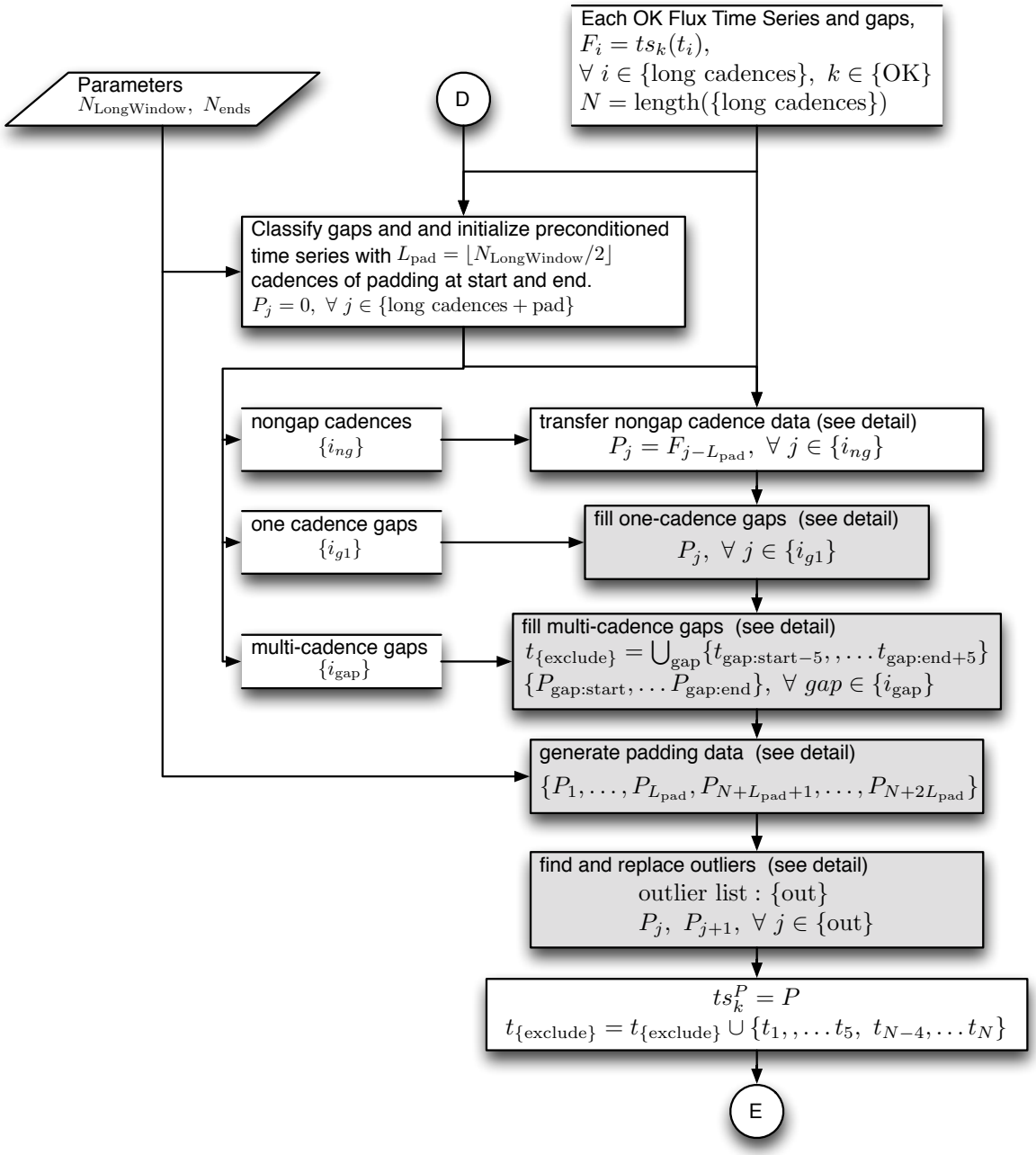


FIGURE 12. **Preconditioning segment (D-E) of SPSD detection algorithm.** The algorithm requires an N-cadence input time series, F, with pre-identified gap locations, and parameters defining the window length in cadences for modeling the time series ends used to generate padding data, N_{ends}, and the window length of the detection filter, $N_{\text{LongWindow}}$. It produces an $N+2\lfloor N_{\text{LongWindow}}/2 \rfloor$ cadence padded, gap-filled and outlier-corrected time series, P and set of cadence times, $t_{\{\text{exclude}\}}$, to exclude as a result of their < 5-cadence proximity to multi-cadence gaps or the time series ends. Processes in shaded boxes represent the distinct algorithms associated with single-cadence gap-filling, multi-cadence gap-filling, end pad generation and outlier detection/correction, and are detailed in following sections. See figs. 10, 33, 34, 35, and 36.

whether they are noise-induced, astrophysical in origin, or SPSDs. (To identify multi-SPSD time series, we simply iterate the process with only with those targets in which SPSDs were previously found.) The evaluation requires that we know what to expect, so before starting we calculate a set of detection thresholds, which depend only on three quantities: the number of long cadences in the time series, N, the length of a window around the maximum to search for an accompanying minimum, suggestive of an astrophysical cause for the maximum, which we set at $N_{\mathrm{LongWindow}}$, and the false positive rate, f_{FP}, which specifies how deeply we are willing to dive into the noise distribution to bring up true SPSDs, at the expense of increasing numbers of false positives. The details of these calculation are provide in Appendix B and fig. 37 therein.

The first step in the process is to produce a set of quantities which characterize the extreme value properties of each detection time series, ts^{D3}. For the kth time series, we determine the following quantities:

- ε_k, the maximum of ts_k^{D3};
- t_k^{ε}, the cadence when the maximum occurred;
- ζ_k, the minimum in a window $\pm\lfloor N_{\mathrm{LongWindow}}/2 \rfloor$ around t_i^{ε}; and
- Σ_k, the sum of ε_k and ζ_i.

Next, the algorithm checks for a companion negative extremum in close proximity to any potential SPSD candidate which would indicate a transiting body, rather than an SPSD. If a comparable minimum is detected, the searched interval is zeroed out like the gap buffers, and the next maximum is calculated, and the process continues until an isolated maximum is found. It is obvious that this process will reduce the detection efficiency in proportion to the number of excluded cadences. Finally, the algorithm collects isolated maxima, ε_k, and extrema sums, $Sigma_i$, for all the targets, applies the SPSD acceptance criteria described below, forming the subset {SPSD candidates}.

The SPSD acceptance criteria requires that the maximum, ε_k, exceed the threshold $u\,(N, f_{\mathrm{FP}})$ and the criteria indicating a astrophysically-induced associated minimum are *not* met. The probability is f_{FP}, that the maximum of a standardized normal distribution with the N samples would be larger than the threshold $u\,(N, f_{\mathrm{FP}})$. The veto criteria, indicating a possible transit, are the same as those used to find the isolated maximum, so this criteria is never met in the currently implemented version of the algorithm. It is a vestige of an earlier version, which only detected SPSDs that were larger than the deepest transit signal in the time series. Nevertheless, since the criteria is now used in the maximum determination described above, we discuss it here.

There are 2 main astrophysical phenomena which might induce a rapid drop in the stellar flux, a transit due to eclipsing binary or planet, or a flare. Either will partner the flux drop with a similar flux increase. In the case of a flare, the the drop is not likely to be instantaneous, and therefore probably less likely to be confused with an SPSD. But, because an SPSD accidentally occurring close to a sudden rise in flux is comparably surprising, we chose to search for a minimum in a window which encompasses both sides of the detection time series maximum. The 2 day window half-width is 4× longer than an earth transit, however again we have weighed the likelihood of astrophysical origin of 2 similar opposite-going events in close temporal proximity against an uncommon SPSD occurring by chance around the time of a sudden rise in detected flux. We exclude a maximum which exceeds the noise criterion above from consideration as an SPSD, if either of the following holds regarding the sum, $Sigma_k$, of the maximum and neighboring minimum:

- $\Sigma_k < u_{\Delta}\,(N, N_{\mathrm{LongWindow}}, f_{\mathrm{FP}})$, where u_{Δ} (see App. B) is the difference threshold for a false positive rate of f_{FP} OR

- $\Sigma_k < R_{\text{min:max}}\varepsilon_k - u\,(N_{\text{LongWindow}}, 0.5)$, where $R_{\text{min:max}}$ is the specified min:max ratio tolerance, and the probability is 50%, that the minimum of a standardized normal distribution with the $N_{\text{LongWindow}}$ samples would be less than the threshhold $u\,(N_{\text{LongWindow}}, 0.5)$

Note that for a perfectly symmetric transit signature, the sum of the maximum and minimum would be zero. The first criterion constrains the accidental occurrence of the observed sum, and the second is on the equivalence of the absolute values of the maximum and minimum. If either the maximum and minimum agree within tolerance, or the probability is less than f_{FP} of the sum occurring by accident, then the possibility that the maximum is due to an SPSD is vetoed. See Appendix B for a more detailed derivation.

Candidate Validation. Fig. 14 is a flow diagram of the validation segment (I-J) of the detection algorithm. The algorithm performs fits of candidate SPSDs using 2 different window lengths and examines the step height obtained in the two cases to verify that the detected feature is negative and statistically significant at both scales and that the ratio of estimated step heights is within a specified tolerance, R_{steps} of each other.

The input data for the fits are segments of the preconditioned time series, $ts^P_{\{\text{candidates}\}}$, where {candidates} is the set of targets with identified SPSD candidates. The segments are centered on the SPSD candidate time, t^{ε} and of length $N_{\text{LongWindow}}$ for the long window fit and, $N_{\text{ShortWindow}}$ for the short window fit. The model design matrices, M and pseudoinverses, A are described in §2.1. For each case, we apply the pseudoinverse to the data segment to produce coefficient vectors $c^{L,J}$, and $c^{S,J}$ where L and S indicate long and short and the J indicates the coefficient applies to the Jth model component. We then apply the coefficients to the model matrix to generate the model estimates, $ts^L_{\text{est,k}}$ and $ts^S_{\text{est,k}}$ (k is a target index). Next, we calculate step heights, h_L and h_S, based on the estimated differences in values ± 2 cadences from the window center. We do this instead of just taking the coefficent corresponding to the step term in the model because it accounts for small contributions of the other terms in the model and avoids the three *delta*-function terms at the center and ± 1 cadence. We want to know whether these step heights are statistically significant so we need to estimate the uncertainty. The shot noise per value is estimated as simply $\sqrt{\overline{F}}$, where $\overline{F}$ is the mean flux in e^-/long cadence. $\overline{F}$ is estimated in the fits by $c^{L,2}$, and $c^{S,2}$ assuming the second component of the model is a constant as defined in fig. 5. Ignoring the 3 central values in a window X, where X is L(long window) or S(short window), the uncertainty in the flux estimate due to shot noise on each side of the step is $\sigma_{\text{shot}} = \sqrt{2c^{X,2}/(N_{\text{X}} - 3)}$, scaling as the inverse of the square root of the number of points in the average, $(N_{\text{X}} - 3)/2$. The shot noise limit on the difference between 2 such values, i.e. a step height measured with such a window, is just $\sqrt{2}\,\sigma_{\text{shot}}$. The SPSD step height is a small fraction of the flux, so we can define a step height standardized to shot noise, $\widehat{h}_{X,k}$, which measures the significance of the step relative to shot noise limit,

$$\widehat{h}_{X,k} = \frac{h_{X,k}}{\sqrt{2}\,\sigma_{\text{shot}}} = \sqrt{\frac{(N_{\text{X}} - 3)h^2_{X,k}}{4c^{X,2}_k}}.$$

This quantity is used to verify that the step height is significant relative to the shot noise limit at both scales, and thereby help to validate that the candidate is an SPSD. For this purpose, we ignore read noise which is not dominant until $\sim$ 16th magnitude.

Although they are not currently used in the validation process, we also calculate two diagnostic quantities regarding the goodness of fit, based on the residuals, $r^X_k = ts^P_k - ts^X_{\text{est,k}}$. We robustly estimate the standard deviation of the residuals as the median absolute deviation of r^X_k using, $\sigma^X_k = \frac{\pi}{2}\mu_{1/2}(\left|r^X_k - \mu_{1/2}(r^X_k)\right|)$ and a noise estimate, η^X_k, from residual first differences, Δr^X_k, using $\eta^X_k = \frac{\pi}{2}\mu_{1/2}(\left|\Delta r^X_k - \mu_{1/2}(\Delta r^X_k)\right|)$. These

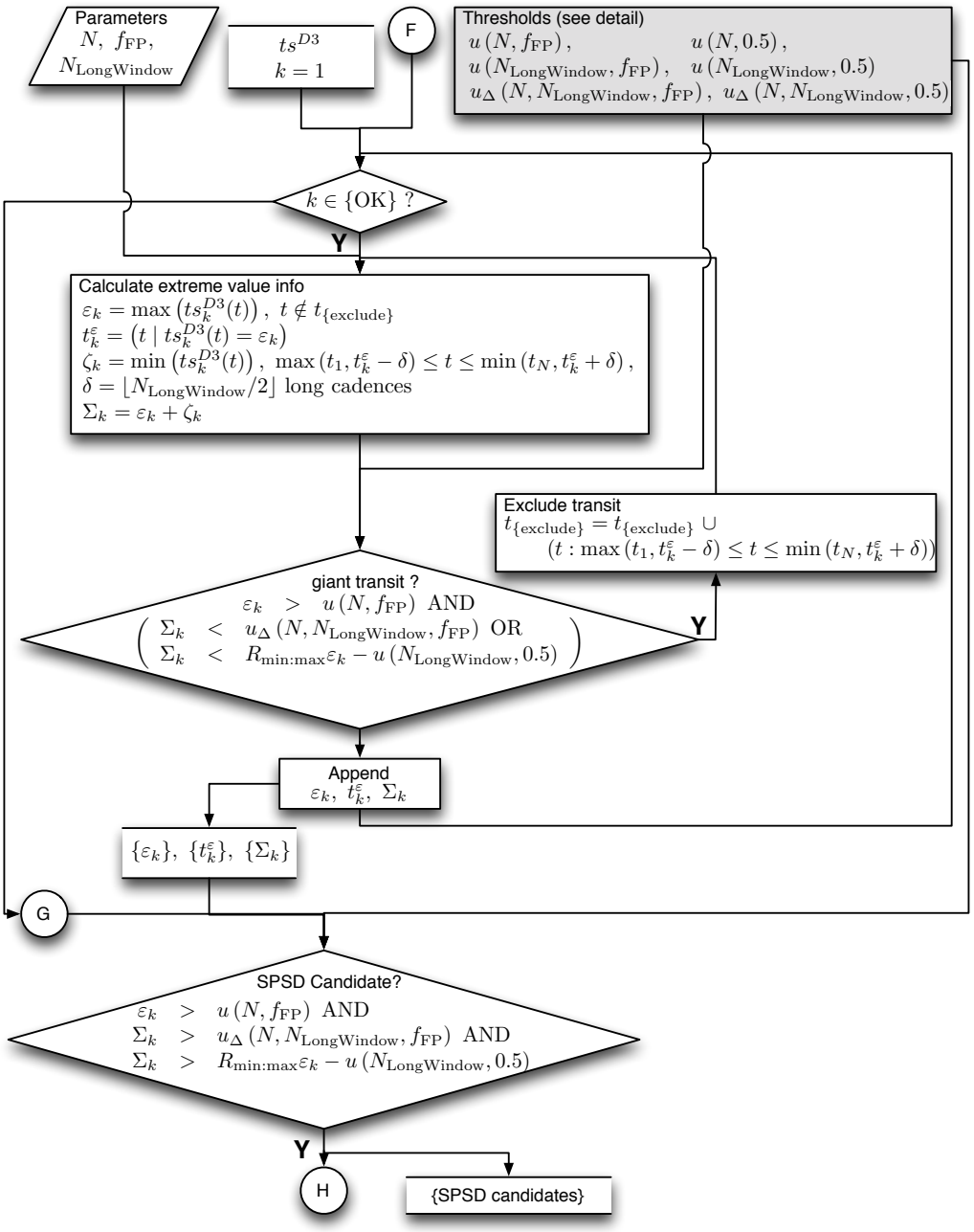


FIGURE 13. **Extreme value and candidate ID segments (F-G-H) of SPSD detection algorithm.** The algorithm requires both the input parameter and the output of the extreme value threshold calculations (shaded box) shown in fig. 37, and a set of standardized detection time series, ts^{D3}. It produces a set of SPSD candidates. The process first calculates a set of quantities which characterize the extreme value properties of the detection time series: ε_k, the maximum of ts_k^{D3}; t_k^ε, the cadence when the maximum occurred; ζ_k, the minimum in a window $\pm\lfloor N_{\rm LongWindow}/2 \rfloor$ around t_k^ε; and Σ_k, the sum of ε_k and ζ_k. It implements a transit veto loop to select only maxima which do not have close-proximity matching minima, and then choose only time series with threshold-crossing maxima as candidate SPSDs. We detect at most one candidate per target in an iteration of this segment. See figs. 10 and 37.

parameters are output as potential downstream V&V diagnostics for both the long window and short window models.

We confirm that a candidate is a valid SPSD on the basis of 5 criteria. The steps must be downward-going in time in both fits, $h_L < 0$ and $h_S < 0$. The steps must be 3× larger than the shot noise limits, $\widehat{h}_L > 3$ and $\widehat{h}_S > 3$. Finally, the ratio of the steps determined by the two models must be within a specified tolerance. To formulate this criterion, we use the quantity $\Lambda = \left|\log\left|\frac{h_L}{h_S}\right|\right|$ (here log is the natural logarithm) because it is always real, positive and symmetric with respect to exchange of h_L and h_S. If Δh is the uncertainty in h, then the uncertainty in Λ is

$$\Delta\Lambda = \sqrt{\frac{\Delta h_L}{h_L}^2 + \frac{\Delta h_S}{h_S}^2} \approx \sqrt{\frac{1}{\widehat{h}_L^2} + \frac{1}{\widehat{h}_S^2}},$$

where we have simple substituted the shot noise limits for $\Delta h_{L,S}$. To account for uncertainty in the criterion we specify the tolerance with respect to the 1σ lower limit on Λ, so $\Lambda - \Delta\Lambda < R_{\rm steps}$. For the default tolerance $R_{\rm steps} = 0.7$, this criterion is perhaps more easily stated: the 1σ lower limit of the ratio of the larger step height to the smaller step height is less than two. The fit coefficients, step heights, and fit diagnostics are output for all confirmed candidates.

2.3. **Correction Algorithm.**

Overview. We assume an input flux time series, $ts(t)$ may be decomposed as

$$ts(t) = T(t)\epsilon_T(t) + B(t)\epsilon_B(t) + \beta(t) + \eta(t)$$

where $T(t)$ is the time varying target flux, $\epsilon_T(t)$ is the time varying aperture throughput for the target, $B(t)$ is any remaining time varying flux from background sources, $\epsilon_B(t)$ is the time varying aperture throughput for this background flux, $\beta(t)$ is any uncorrected time varying offset from the flux values, which may also be viewed as correlated noise, and $\eta(t)$ is the uncorrelated white noise. From the viewpoint of signal processing $\beta(t)+\eta(t)$ may be combined into a colored noise component, however from the instrument design perspective they are distinct. $\beta(t)$ can be further broken into a correctable part, $\beta_c(t)$, and an uncorrectable part, $\beta_u(t)$, and it makes more sense to call $\beta_u(t)+\eta(t)$ the colored noise component. We will not delve into what makes something correctable or uncorrectable, but note that correlation with other observables or having a unique time signature enable correction. We would like to also account for SPSDs in this picture, but the SPSD might impact either the ϵ factors or the $\beta_c(t)$ term, where we optimistically assume that they are correctable. We further decompose $\epsilon_T = \epsilon_{T0}(1 + \delta^T_{\rm SPSD})$, $\epsilon_B = \epsilon_{B0}(1 + \delta^B_{\rm SPSD})$, and $\beta_c = \beta_{c0} + \beta_{\rm SPSD}$, with implicit definitions. Note that typically $(X(t) - \overline{X})/X(t) \ll 1$ and $(\epsilon_X(t) - \overline{\epsilon}_X)/\epsilon_X(t) \ll 1$), where $X \in \{T, B\}$ and $\{\overline{X}, \overline{\epsilon}_X\}$ are time averages of $\{X, \epsilon_X\}$, so we assume we can ignore terms $(X(t) - \overline{X})(\epsilon_X(t) - \overline{\epsilon}_X)$ in comparison to $\overline{X}(\epsilon_X(t) - \overline{\epsilon}_X)$ or $(X(t) - \overline{X})\overline{\epsilon}_X$. Because *Kepler* is concerned with variations at the 10^{-5} level, this assumption is not always valid, nevertheless, it is pervasive throughout PDC. Taking all this into account we rearrange into

$$\begin{array}{rcll} ts(t) & = & (T(t)\overline{\epsilon}_X + B(t)\overline{\epsilon}_B) & (= \text{flux}) \\ & + & (\overline{T}(\epsilon_{T0}(t) - \overline{\epsilon}_T) + \overline{B}(\epsilon_{B0}(t) - \overline{\epsilon}_B) + \beta_{c0}(t)) & (= \text{systematics}) \\ & + & (\overline{T}\overline{\epsilon}_T\delta^T_{\rm SPSD}(t) + \overline{B}\overline{\epsilon}_B\delta^B_{\rm SPSD}(t) + \beta_{\rm SPSD}(t)) & (= \text{SPSD}) \\ & + & (\beta_u(t) + \eta(t)) & (= \text{noise}), \end{array}$$

where we have also assumed $\epsilon_{X0}(t)\delta^X_{\rm SPSD}(t) = \overline{\epsilon}_X\delta^X_{\rm SPSD}(t) + (\epsilon_{X0}(t) - \overline{\epsilon}_X)\delta^X_{\rm SPSD}(t) \cong \overline{\epsilon}_X\delta^X_{\rm SPSD}(t)$, because, like $(\epsilon_{X0}(t) - \overline{\epsilon}_X)/\overline{\epsilon}_X$, $\delta^X_{\rm SPSD}(t)/\overline{\epsilon}_X \ll 1$.

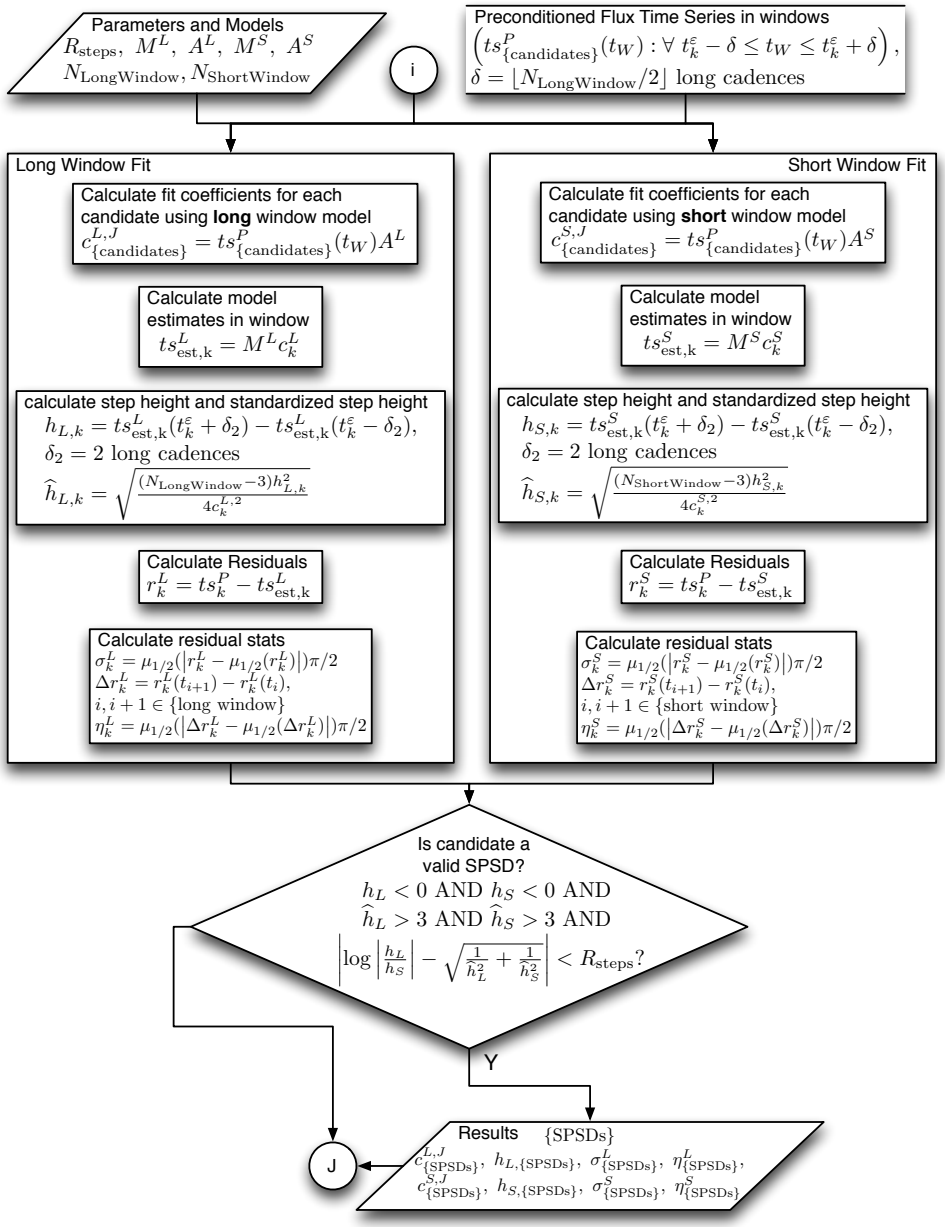


FIGURE 14. **SPSD validation segment (I-J) of SPSD detection algorithm.** The algorithm requires both the input parameter and the output of the detection filter derivation algorithm shown in fig. 5, as well as an additional validation parameter, and a set of preconditioned flux time series from SPSD candidates, $ts^P_{\{\rm candidates\}}$. The validation parameter is $R_{\rm steps}$, a tolerance for the ratio of steps measured using the long:short window models. It produces a set of identified SPSDs and associated long and short model fit coefficients, c, step heights, h, residual σ (derived for median absolute deviation), and residual noise, eta. The validation is based on results of model fits in a short and long window around the candidate. To be a valid SPSD, the models must indicate negative step heights (h) and a 3σ detection above the shot noise limits ($\widehat{h}$) for both scales. The ratio of the step heights h_L/h_S must also be within the specified tolerance after accounting for uncertainties. See fig. 4.

The purpose of the correction algorithm is to extract the SPSD term shown above from the input flux time series as a correction term. Our SPSD model has two main components, a persistent step function and a recovery function, and correction is a two stage process. The first stage estimates a persistent step height from analysis of the entire flux time series, excluding a short recovery window following the SPSD, which typically contains a transient signal. The second stage models the recovery window using a series of exponentials of varying time constant. The algorithm also detects and preserves sinusoids in the signal while removing only the step and recovery transient components. The persistent step correction is refined in each stage in the process. Fig. 15 is a flow diagram of the correction algorithm described below. The shaded boxes are further diagramed in figs. 18 19, and 21, with some details relegated to Appendix C.

Parameters. The input correction parameters specify various aspects of the models used in the correction process. Table 6 lists the parameters and cross-references with variable names in the code. The window lengths define modeled regions of interest used in all correction stages. The polynomial order is only used at the first stage which provides an initial estimate of the persistent step height. The false positive rate is for the sinusoid detection algorithm, specifying the signal above the amplitude spectrum noise level required for inclusion in the model. The last three parameters specify the start, end, and increment of a series of exponential time constants (τ) corresponding to the exponential basis elements used to model the SPSD recovery. These are specified logarithmically because linear specification of time constants often results in a basis of exponential functions which are too similar to each other, i.e., too highly correlated, and which tend to degenerate further when combined with the polynomial nuisance components of the model.

Parameter	**Description**	**Default**	`MATLAB` **Name**
N_W	Half-width of close-proximity modeling window in long cadences	480	`.polyWindowHalfWidth`
N_R	Width of recovery window following SPSD in long cadences	240	`.recoveryWindowWidth`
n_p^{BP}	The polynomial order used in the big picture model	6	`.bigPicturePolyOrder`
f_{FP}^{H}	The specified rate of false positive sinusoids due to noise	0.01	`.harmonicFalsePositiveRate`
$\log_{10}\tau_{\min}$	Minimum value of $\log_{10}$(time constant) for exponential terms in the recovery mode	-2.0	`.logTimeConstantStartValue`
$\log_{10}\tau_{\max}$	Maximum value of $\log_{10}$(time constant) for exponential terms in the recovery model	0.0	`.logTimeConstantMaxValue`
$\Delta\log_{10}\tau$	Increment of $\log_{10}$(time constant) for exponential terms in the recovery model	1.0	`.logTimeConstantIncrement`

TABLE 6. **Correction Parameters.** These parameters specify SPSD correction models. The false positive rate is used to calculate detection thresholds for identifying sinusoidal signals in the flux time series (see Appendix C).

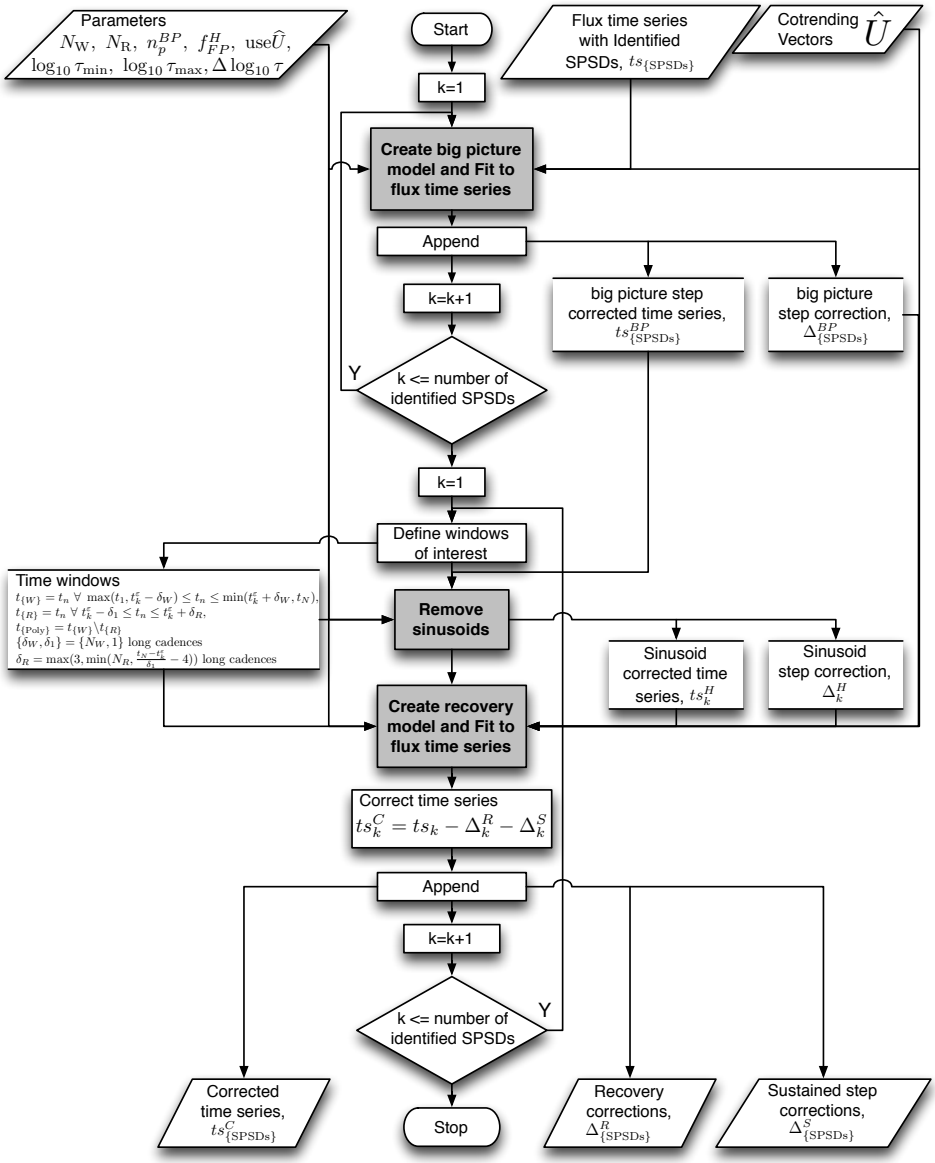


FIGURE 15. **SPSD correction algorithm.** The algorithm requires a set of correction parameters, and a set of uncorrected SPSD-containing flux time series, $ts_{\{\rm SPSDs\}}$ and associated occurrence times, $t^\epsilon_{\{\rm SPSDs\}}$. As implemented, ts have been partially corrected with a coarse PDC-MAP pass [2] and the resulting co-trending basis vectors (CBV), $\widehat{U}$, are also used by the algorithm. The correction parameters are: the window half length in cadences for modeling the time series in close proximity to the SPSD, $N_{\rm W}$; the window length in cadences for modeling the recovery following occurrence of the SPSD, $N_{\rm R}$; the polynomial order for modeling the time series distant from the SPSD, $n_{\rm p}^{BP}$; the accepted fractional rate of false positive sinusoid detections, $f_{\rm FP}^H$; a set of parameters used to specify the basis for modeling the recovery interval, which define a set of exponential time constants, τ, in terms of the base-10 logarithm of the minimum value, maximum value, and difference between consecutive values. τ is specified as a fraction of the full recovery window length. It produces sustained step corrections, recovery corrections, and corrected time series for each input time series. Processes in shaded boxes are detailed in following sections. See figs. 4, 18, 19, and 21.

Input/Output. The inputs for the algorithm include flux time series with detected but not yet corrected SPSDs, the gap indicators and corresponding SPSD occurrence times of these, and the coarse PDC-MAP co-trending basis vectors (CBV), as listed in Table 7. Each time the algorithm is called, only one SPSD from each time series is known and corrected. The algorithm's principle outputs are the 2 components of the calculated correction: the sustained step and the recovery, and a corrected flux time series to which these have been applied. See Table 8.

Input Variable	**Description**	`MATLAB` **Name**
$ts_{\{\text{SPSDs}\}}$	SPSD-containing uncorrected time series	`.timeSeriesStruct.fluxResiduals(iDedStruct.index,:)`
$g_{\{\text{SPSDs}\}}$	gap indicators for time series	`.timeSeriesStruct.gaps(iDedStruct.index,:)`
$t^{\varepsilon}_{\{\text{SPSDs}\}}$	cadence-of-occurrence for each SPSD	`iDedStruct.spsdCadence`
$\widehat{U}$	coarse PDC-MAP co-trending basis vectors	`.timeSeriesStruct.parameters.U_hat`

TABLE 7. **Correction Algorithm Inputs.** The algorithm requires the index set, {SPSDs}, to SPSD-containing flux time series, the occurrence times, $t^{\varepsilon}_{\{\text{SPSDs}\}}$ and the CBV, $\widehat{U}$.

Output Variable	**Description**	`MATLAB` **Name**
$\Delta^{S}_{\{\text{SPSDs}\}}$	persistent step correction for each time series	`..persistentStep`
$\Delta^{R}_{\{\text{SPSDs}\}}$	recovery correction for each time series	`.recoveryTerm`
$ts^{C}_{\{\text{SPSDs}\}}$	corrected flux time series	`.correctedTimeSeries`

TABLE 8. **Correction Algorithm Outputs.** The algorithm outputs correction time series for both a persistent step and a recovery function, Δ^S and Δ^R, and corrected flux time series $ts^C = ts - \Delta^S - \Delta^R$.

Description. The three main stages in the algorithm shown in fig. 15 occur serially and only once per call. (We use the word "stage" instead of "step" to avoid confusion with one of the correction components which is a step function.) The purpose of the various stages is to efficiently arrive at good estimates for the corrections in the presence of a wide variety of stellar behaviors. The empirically derived SPSD model consists of two components:

a persistent step function: This term accounts for the observation that in many cases the measured flux in quiet stars never returns completely to values detected prior to the SPSD. We assume that this is caused by permanent radiation damage in the photosensitive region of the CCD pixels which results in permanently reduced quantum efficiency. As a result, we constrain the step height to a value ≤ 0.

a recovery function: As discussed above, if an SPSD occurs at t_i, then in most cases an exponential transient recovery period begins at t_i. The recovery function described below effectively models this transient behavior. Presumably, this transient behavior results from temporary radiation damage

in the photosensitive region of the CCD pixels, which heals with a time constant which depends on some unobserved characteristics of the damage.

Fig. 16 illustrates the effect of these two terms on a flux time series.

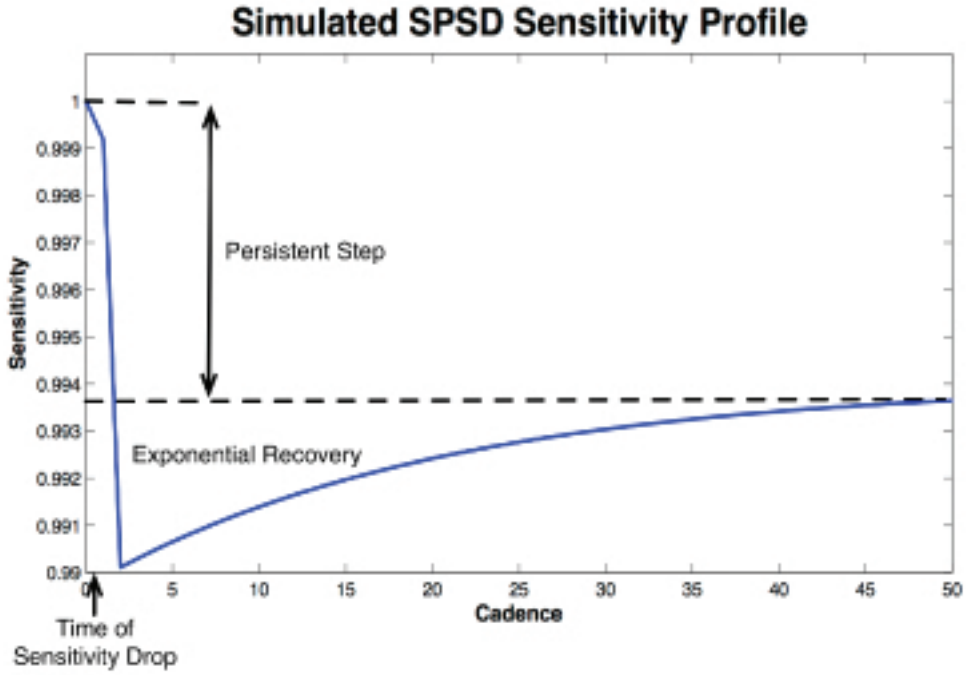


FIGURE 16. **SPSD model.** Our SPSD model consists of an instantaneous sensitivity drop, an exponential partial recovery of sensitivity, and a persistent sensitivity differential. This model both informed the algorithms design and provided a basis for simulation testing.

We found in the development process that attempts to fit all the necessary components of the whole light curve at once required a large basis. This often resulted in poor results because the basis needed to fit the stellar behavior, either also fit a large fraction of the SPSD effects, or was highly degenerate with the SPSD model terms and resulted in nonsensical model coefficients. Instead, we adopted a three-stage approach where we operate on specific time segments or windows of the time series at each stage to extract incrementally better estimates of the persistent step at each stage while removing components that confound recovery determination. These three stages are:

big picture modeling: Here we extract an initial estimate of the persistent step by modeling the light curve, excluding the recovery-encompassing period after the SPSD. The model is a step, a polynomial of order n_p^{BP}, and the CBV. The resulting coefficient of the step component is the initial estimate of the persistent step height. The details of this segment of the algorithm are discussed below and diagramed in fig. 18.

sinusoid removal: Here we detect the presence of sinusoidal features in the big picture step-corrected time series, and if detected, model a region in close proximity before and after the SPSD occurrence time. The model includes another step and a series of sinusoids as described below, in fig. 19 and fig. 39 in Appendix C. The resulting coefficient of the step component is an incremental additive

adjustment to the persistent step height. The flux time series with both steps and the sinusoids removed is used in the next processing stage.

recovery modeling: Here we extract and estimate of the recovery function by modeling a region in close proximity before and after the SPSD occurrence time. The model includes a step, a polynomial series, a series of exponential functions, 3 *delta*-functions, and the CBV as described below and in fig. 21. The order of the polynomial is determined by a preliminary process from data in this close proximity window, but excluding the recovery time interval. The process uses Akaike's Information criterion to select the optimal polynomial order. The resulting coefficient of the step component is another incremental additive adjustment to obtain the final persistent step height correction. The combination of the series of exponential functions and 3 *delta*-functions represent the final recovery correction.

Fig. 17 illustrates the regions of interest, relationships and products of these stages. The combined process provides satisfactory results on a much larger fraction of light curves than simpler approaches that we have tried. We found that $\sim 10\%$ of light curves have a sinusoidal component with period greater than the detection filter cutoff at 70 long cadences. Many more have rather complex low frequency or red noise components at periods longer than 300 long cadences. We found in trying to implement these simpler models that a number of difficulties arose:

- the sustained step was often biased,
- the recovery was partially removed by the large number of components required to model the wide variety of stellar behaviors leading to an underestimate of the recovery,
- the recovery model components were often highly degenerate with the remaining components leading to an unnaturally large recovery correction and corresponding opposite distortion in the corrected light curve,
- the fit was slow due to the large number of time series values and large number of components required for the model,
- the presence of sinusoidal signals in the light curve required nonlinear fitting of multiple frequencies in the data, which was slow, and it was difficult to guarantee convergence without a fairly complex implementation,
- we don't know in advance which harmonics or subharmonics of a detected frequency are needed to model the full periodic waveform, so we must either include many or implement a complex detection algorithm,
- the sinusoidal signals were often time varying, requiring additional nuisance model parameters which exacerbate many of the problems listed above.

In the context of determining the SPSD corrections, the model parameters describing the stellar and systematic variations in the light curve are nuisance parameters. In fact for the default input parameters, even though we fit for dozens of model coefficients, only 9 are actually used in the corrections. We fit for the others in the order described here simply to get reasonable estimates for these 9. The paragraphs below describe the three stages in more detail and explain how they help resolve the problems listed above.

Big Picture Modeling. We named this "big picture" modeling because, conceptually, we stand back and look at the big picture to derive a first estimate the sustained step. The parameter of interest is the step coefficient. The polynomial, and CBV terms are nuisance parameters. We exclude the recovery window because the recovery does not constrain the *sustained* step, but instead the instantaneous step, which is the sum of the sustained step and the additional instantaneous step we include in the recovery as shown

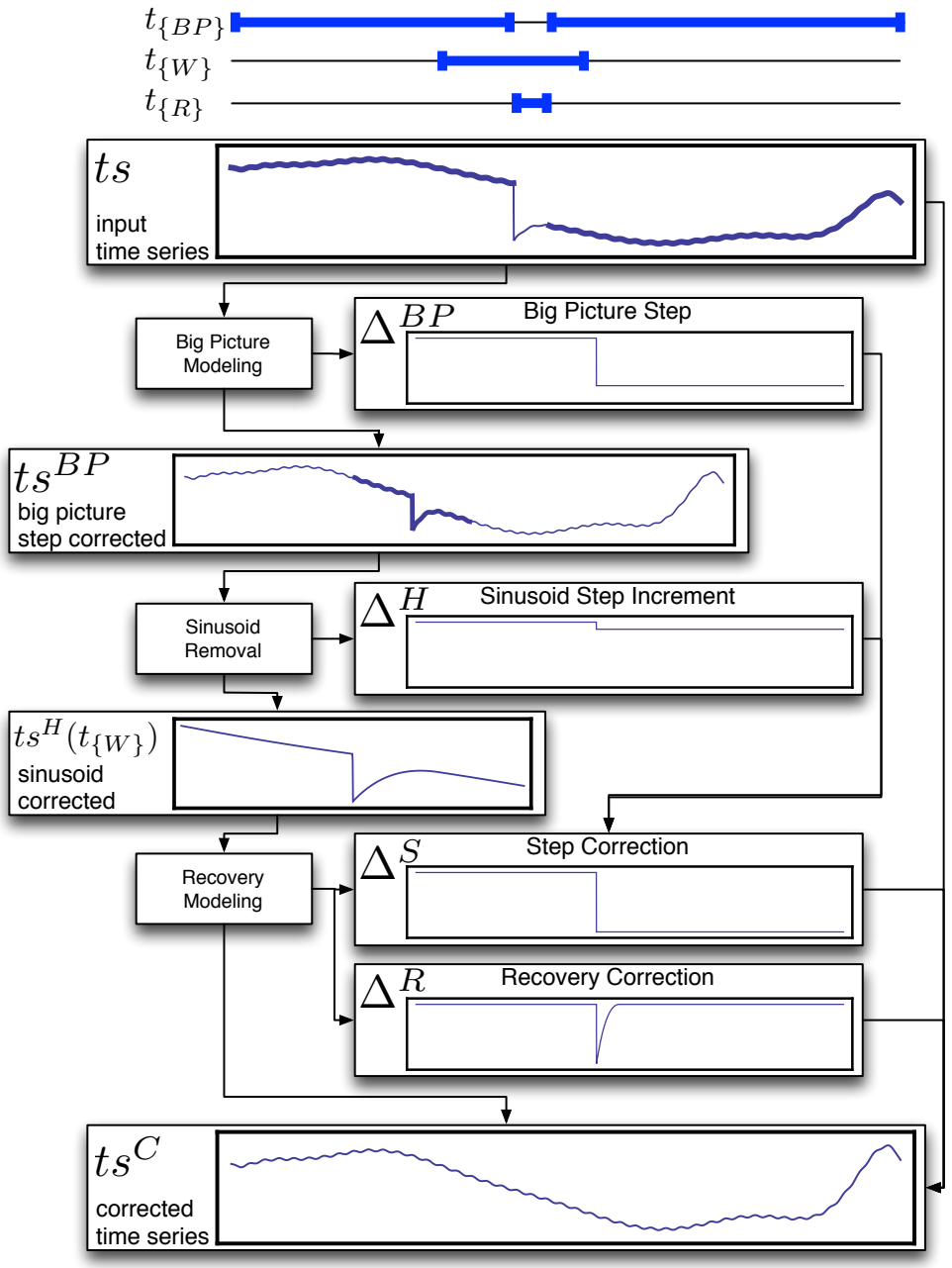


FIGURE 17. **SPSD correction algorithm roadmap.** An input flux time series, ts, undergoes big picture modeling over a region of interest which excludes $t_{\{R\}}$, producing a big picture step correction, Δ^{BP}, and big picture step corrected time series, ts^{BP}. ts^{BP} then undergoes sinusoid removal in the region of interest, $t_{\{W\}}$, to produce an increment to the step correction, Δ^{H}, and a sinusoid-corrected time series, ts^{H}. Recovery modeling is then performed on $ts^{H}(t_{\{W\}})$, to produce the recovery correction, Δ^{R}, and the final step correction, Δ^{S} from the sum of Δ^{BP}, Δ^{H} and an incremental step adjustment from the recovery modeling. The SPSD-corrected time series is $ts^{C} = ts - \Delta^{S} - \Delta^{R}$. The big picture window, $t_{\{BP\}}$, close proximity window, $t_{\{W\}}$, and the recovery window, $t_{\{R\}}$, are indicated for ts by the blue intervals at the top.

in fig. 16. By excluding this time interval at the first stage, we avoid introducing any bias in the step coefficient due to the recovery, and we don't need to include the recovery terms and all the high frequency terms which are needed to accurately model the non-SPSD behavior in order to accurately estimate the recovery. Polynomials are used because a smaller basis will fit the partial low frequency sinusoids which are often encountered in the single quarter light curves in comparison to using sinusoidal functions. A partial sinusoid produces a continuum in its Fourier spectrum and therefore requires a large basis of sinusoidal functions for linear modeling.

We find that this approach works well even if we significantly underrepresent any higher frequency components that may exist in the light curve, because these high frequencies tend to average over the typically long time intervals of interest. We found satisfactory results by selecting a relatively low order, nominally 6, for the polynomial component. The 6th order polynomial crosses zero roughly every 2 weeks for quarterly processing, which is a $\sim 3\times$ the nominal 5 day length of the recovery window. If the time between zero crossing gets much closer than this to the length of the model gap due to the excluded recovery interval, then the polynomial terms can sum to introduce a step degeneracy which would bias our step estimate. To summarize, we model the stellar behavior with low order polynomials, the systematic behavior with the CBV, and exclude the recovery to get a reasonably unbiased initial estimate of the sustained step caused by the SPSD. We find that we can leave the high frequency stellar behavior unmodeled at this stage without harming the step estimate. An exception is when the SPSD occurs near the beginning or end of the time series, where we are unavoidably limited in our knowledge of the stellar behavior either before or after the step. In these cases it is often more difficult to determine, even by eye, the degree to which the step estimate is biased.

Fig. 18 is a flow diagram of the big picture modeling segment of the algorithm. We model the input flux time series, ts_k, with SPSD identified at time, t_k^ε, in a time window, $t_{\{BP\}}$, where $\{BP\}$ is the included set of cadence indices and k is a target index in {SPSDs}. $\{BP\}$ includes all indices except data gaps and the *recovery gap*, which is defined as the closed interval from 1 cadence before t_k^ε to N_ρ cadences after t_k^ε, where $N_\rho = \min(N_R + 1, \frac{t_N - t_k^\varepsilon}{\delta_1} - 4)$, with t_N, the time of the last long cadence; δ_1, the duration of a long cadence; and N_R, the input parameter specifying the length of the recovery window. The length N_ρ assures at least 4 cadences after the recovery gap for determining the step coefficient when the SPSD is near the time series end. The 5-cadence detection exclusion zone (see § 2.2) at the beginning of the time series ensures at least 4 cadences are included before the recovery gap for determining the step coefficient when the SPSD is near the time series start.

The linear model includes $n_p^{BP} + m + 2$ terms, where n_p^{BP} is the input parameter specifying the big picture polynomial order, and m is a count of CBV provided by coarse PDC-MAP. The 2 additional terms are a constant (= 1 for all included cadences) and a step (= 0 before t_k^ε, and = 1 after t_k^ε). The polynomial terms are represented by the Legendre polynomials $P_j(x)$ for j ranging from 1 to n_p^{BP}. Here x, which ranges from -1 to 1, is a scaled time variable determined from the cadence index, i, by $x_i = 2\frac{i-1}{N-1} - 1$, where i is an index in $\{BP\}$ which will range between 1 and N. The least-squares fit results in a set of coefficients c_k^J with a J corresponding to each term. The J corresponding to the step term ($J = 1$ in fig. 18) is our initial estimate of the sustained step height. The resulting correction is then defined as $\Delta_k^{BP} = c_k^1 \, \Theta(t, t_k^\varepsilon)$, where

$$\Theta(t, t') = \left\{ \begin{array}{ccc} 0 & t & < & t' \\ 0.5 & t & = & t' \\ 1 & t & > & t' \end{array} \right\}.$$

The resulting big picture corrected time series is $ts_k^{BP} = ts_k - \Delta_k^{BP}$.

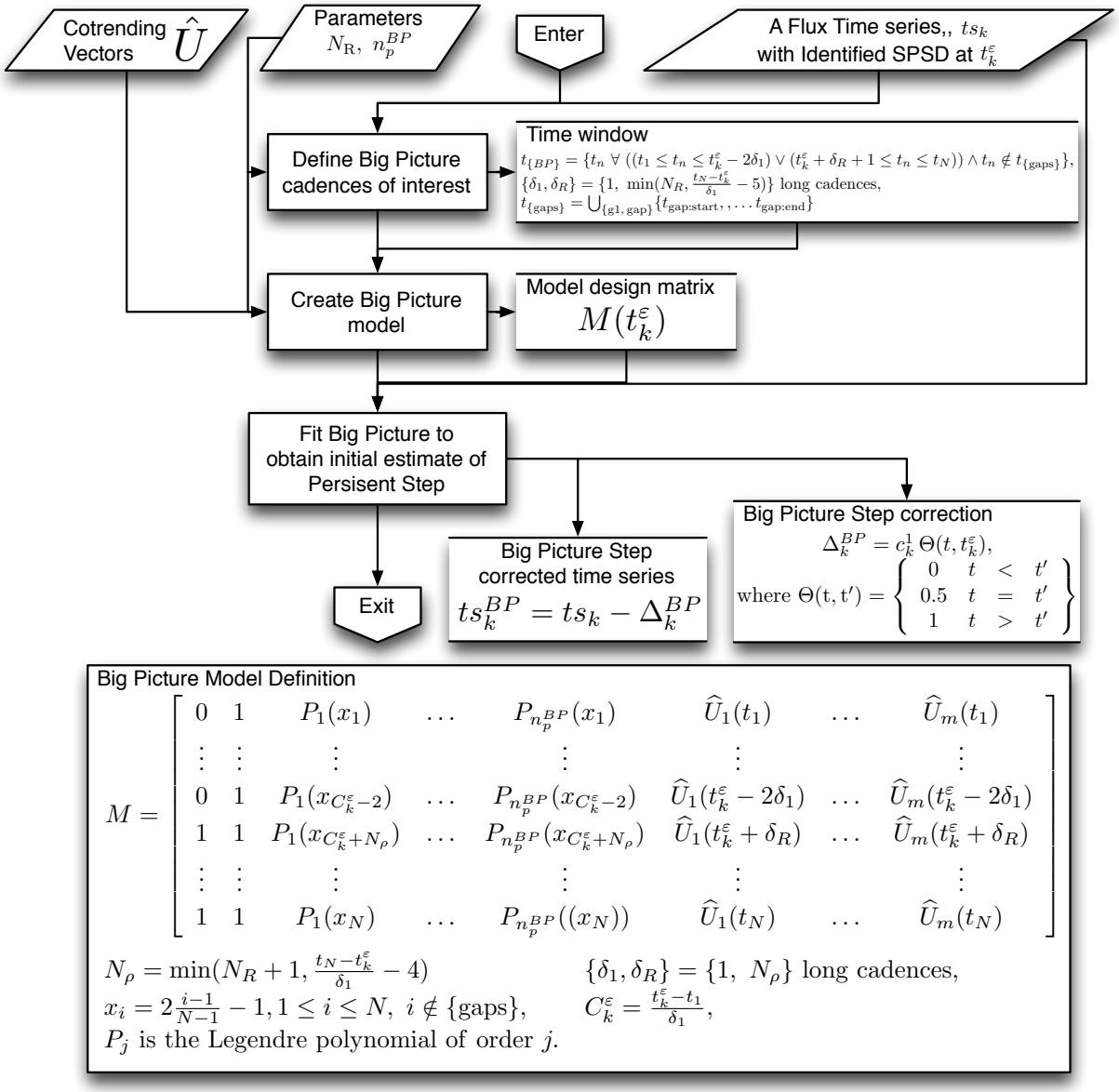


FIGURE 18. **Big picture modeling segment of SPSD correction algorithm.** This segment requires one of the flux time series, ts_k to be corrected along with its SPSD occurrence time, t_k^{ε}, the coarse MAP CBVs, $\widehat{U}$, and the parameter defining the rate of false positive sinusoid detections, f_{FP}^H. The algorithm models the entire flux time series, i.e the *big picture*, excluding the recovery window, to obtain an initial estimate of the sustained step component of the correction. The model includes a step, a constant, Legendre polynomial components, and the CBVs, but all except the step are nuisance parameters, and ignored in later steps. The best-fit step component, Δ_k^{BP} and and the flux time series corrected by the step, ts_k^{BP}, are the only outputs. See fig. 15.

Sinusoid Removal. In development, it was planned that the SPSD detection would follow a harmonic removal algorithm, so we almost decided to ignore issues which arose in light curves with periodic behavior. Concern that the harmonic removal might be incomplete in the presence of SPSDs and the detection of a small but significant number of SPSDs in light curves with periodic behavior, prompted us to develop a joint step-sinusoid removal algorithm. This was a fortunate decision since it gave the flexibility to reverse the order of the SPSD correction and stand-alone harmonic removal modules, which was deemed necessary as PDC-MAP was developed. The algorithm takes a rather brute-force approach since the number of light curves where it is needed is small, usually < 20 per module output per quarter. If the final order of SPSD detection and harmonic removal algorithms had been known, prior to development, we may have spent somewhat more time searching for a more efficient nonlinear technique than the approach discussed below, e.g. a Markov Chain Monte Carlo algorithm.

The parameter of main interest is again the step coefficient which is applied as an adjustment to the initial big picture estimate. The sinusoid harmonic and subharmonic corrections are applied to produce the output sinusoid-corrected time series, but this is only used by the recovery modeling segment and is not a part of the SPSD corrections or corrected time series produced as final output of the overall algorithm. Thus, these too are nuisance parameters. Fitting sinusoids requires frequencies as model parameters, and these are nonlinear. Furthermore, the frequencies are often changing slowly with time over a quarter so the rate of change is also a model parameter, and we don't know how many sinusoids to model, so this number is also a free parameter. Our initial attempts to simultaneously fit the whole time series to numerous sinusoids and the recovery, and other terms had were unsuccessful. The selected approach has features specifically designed to address the identified problems and these are discussed below.

The big picture modeling extracts all the SPSD correction information we need to know about data far from the SPSD occurrence time, so in subsequent steps we are able to focus primarily on data in close-proximity to the SPSD initial step. This smaller data set can be fit much faster, and since there is much less time for a time-varying frequency to change, we can model the frequencies as constant. We also need fewer model components since we can ignore those with long periods compared to the window length. We only use full time series in the detection segment, where we search for peaks in the FFT of the full partially-corrected time series to identify candidate frequencies.

By obtaining initial frequency estimates from the FFT we can use a series of linear fits where we scan over a narrow frequency range to refine the frequency to the level needed for the next stage, thereby avoiding the risk of non-convergent nonlinear modeling. The scanning approach is not the most efficient but because we are working with a time series of reduced length, and we are simultaneously fitting many harmonics and subharmonics of a given base frequency, performance has not been an issue.

In some cases, many sinusoids are modeled and including these along with the exponential recovery terms, CBVs and polynomials which model the non-periodic part of the windowed time series at the next stage would occasionally produced internal degeneracies the model basis which produce nonsensical coefficients and inaccurate SPSD corrections. This is a result of degeneracy with the model components and not the time series data. By fitting the sinusoids separately we do not need to limit ourselves to those harmonics which have significant amplitudes, we can fit all harmonics and subharmonics of the base frequency in a given range of interest. The result might be improved slightly by excluding the recovery range from the close-proximity modeled range at this stage, however results are satisfactory without this exclusion, so it has not been implemented.

To summarize, in this stage we model only the periodic component of the stellar flux, if any, with a basis set of sinusoids which are harmonics and subharmonics of identified base frequencies, and a step component

for the SPSD. The step represents a refinement of the step determined in the big picture modeling stage. The sinusoid modeling occurs in a close-proximity window around the SPSD occurrence time.

Fig. 19 is a flow diagram of the sinusoid removal segment of the algorithm. We model the big picture-corrected flux time series, ts_k^{BP}, with SPSD identified at time, t_k^{ε}, in a time window, $t_{\{W\}}$, where $\{W\}$ is the included set of cadence indices and k is a target index in {SPSDs} as before. $\{W\}$ includes indices in close proximity to the index of the cadence when the SPSD occurred, excluding data gaps. Close proximity is defined as the closed interval from N_W cadence before t_k^{ε}, or the cadence 1 if greater, to N_W cadences after t_k^{ε}, or cadence N if smaller, where N_W, the input parameter specifying the half-length of the close-proximity window. The default close-proximity window half-width is chosen to be twice the recovery window width or roughly 10 days, to assure sufficient data after the recovery to constrain the refinements to the sustained step which occur in both the sinusoid removal and recovery modeling stages.

Prior to modeling, we identify a set of candidate frequencies, $\nu_{\{\text{detected}\}}$, based on an FFT of the full input time series which we rename, ts_k^H. We also initialize the sinusoid step correction, Δ_k^H to zeros. The algorithm segment which does this is described in Appendix C and diagramed in fig. 38. We select the detected frequency with the maximum detection statistic as the *base* frequency, $\nu_{\max}$, for our first modeling pass. The FFT frequency resolution is limited to $\delta\nu = 1/(N\delta_1)$ Hz, where δ_1 is the cadence duration in seconds and N is the time series length, and this is not precise enough to effectively remove the sinusoidal features, so the next step is a series of refinements.

In each base frequency refinement step we construct a set of n_ν models, where each model has a different base frequency, ν_X, and the base frequencies span the range $[\nu_X - \Delta\nu_-,\ \nu_X + \Delta\nu_+]$. Each model includes a range of harmonics and subharmonics of the base frequency and a step component as described in Appendix C and diagramed in fig. 39. Once constructed, we perform a least-squares fit of each model to windowed time series data, $ts_k^H(t_{\{W\}})$ and calculate a robust standard deviation, σ^H, of the fit residuals using the median absolute deviation. The refined frequency, ν_Y is the base frequency corresponding to the model with the smallest value for σ^H. We use this scanning method because it is not unusual to have multiple local minima when modeling frequencies, and we desire the best fit in the range. Also this approach is deterministic, without the risk of divergent results as often happens in packaged nonlinear fitting routines.

This frequency refinement process is represented by the dark gray boxes in fig. 19, where it is applied 7 times. The first time we take $n_\nu = 26$, $\nu_X = nu_{\max}$, $\Delta\nu_- = \delta\nu$, and $\Delta\nu_+ = 2\delta\nu$ to obtain a refined frequency, ν_1. The reason for the this range is that the frequencies are often slightly time varying over the full interval so the spectral feature may be broadened and the peak frequency in the detection FFT may not be the peak in our time window. The range is asymmetric because our detection statistic is the first difference of the amplitudes and and the maximum first difference of a broadened feature may occur slightly before the peak frequency. Next, we define $\Delta\nu_1 = 3\delta\nu/25$ and take $n_\nu = 21$, $\nu_X = nu_1$, $\Delta\nu_- = \Delta\nu_+ = delta\nu_1$ to obtain a refined frequency, ν_2. Subsequently, for j ranging from 2 to 6, we define $\Delta\nu_j = \Delta\nu_{j-1}/10$ and take $n_\nu = 21$, $\nu_X = nu_j$, $\Delta\nu_- = \Delta\nu_+ = delta\nu_j$ to obtain a refined frequency, ν_{j+1}. So, if a sinusoidal signal is detected, the algorithm constructs 152 models, each containing numerous subharmonics and harmonics, in order to refine the base frequency from $nu_{\max}$ to ν_7. This may see like a large amount of processing, but if the periodic signal has sharp features, which correspond to strong high harmonics, then this level of precision is necessary to effectively remove the periodic waveform.

Upon completion of the refinement process the model corresponding to ν_7 is fit to $ts_k^H(t_{\{W\}})$, to obtain coefficients c^H. We the calculate sinusoid correction estimates for the entire time series $\delta\, ts^H = \sum_j c_j^H M_j^H(\nu_7, t_k^{\varepsilon}, t)$, and incrementally apply the correction, $ts_k^H = ts_k^H - \delta\, ts^H$, removing the sinusoids from the time series. Assuming the first component of the model is the step, we also incremental adjust the

sinusoid step correction, $\Delta_k^H = \Delta_k^H + c_1^H M_1^H(\nu_7, t_k^\varepsilon, t)$. The entire process of detection via FFT, refinement and incremental correction is repeated for up to 2 more base frequencies, if detected, to produce final results: the sinusoid step correction, Δ_k^H and sinusoid corrected time series, ts_k^H. Note that if no sinusoids are detected, the output is $ts_k^H = ts_k^{BP}$ and Δ_k^H is a zero vector.

Recovery Modeling. The recovery is modeled with a series of modified exponential functions specified by the recovery window length, N_R and the 3 logarithmic parameters defining the fractional time constant, τ. The parameters of interest are the coefficients of these terms along with those of the three *delta*-functions at the SPSD and ±1 cadence, and if found to improve the fit, a final adjustment to the sustained step term. The model encompasses the same window as the sinusoid model and includes polynomials to account for any remaining non-periodic stellar variation, and CBV to account for systematics.

Selecting a polynomial order that is too high increases risk that these terms bias our estimates of the recovery coefficients either by accounting for a significant fraction of the time series recovery themselves, which produces an underestimate of the recovery term coefficients, or by exhibiting a strong degree of degeneracy with the recovery model terms, which results in unpredictable, and often nonsensical recovery term coefficients. To select a polynomial order which balances good representation of the underlying stellar behavior while limited the risk of suffering from these problems, we use the MATLAB library function `polydeg` [10] to model a set of cadences, {Poly}, which spans the close-proximity window used for sinusoid modeling, but excludes the recovery window. The function `polydeg` provides the optimal polynomial order for fitting the selected data based on Akaike's Information criterion (AIC) , which we interpret as the minimum order which well represents the stellar behavior.

The interval very near (±1 cadence) the SPSD occurrence time is unpredictable. We imagine SPSD occurs instantaneously at one of 270 frames which are accumulated to produce the long cadence data, so the ratio of time at the top or bottom of the step discontinuity is unobserved and thus requires a free parameter in the model. The cadence where the SPSD occurs could include the initiating cosmic ray, in which case the summed value may be above the top level of the step discontinuity and the detection occurs at the following cadence and the last cadence before sometimes contains a cosmic ray which may have been inadequately removed because the nearby SPSD fools the detection algorithm. It also appears that there is occasionally a very short term change in the first cadence after the SPSD. Since the range is so small, we allocate three *delta*-functions in the recovery model, one centered at each of these cadences. The *delta*-functions prevent these cadence data from affecting any of the other model coefficients and they ensure that we avoid residual outliers in these cadences after the recovery correction.

The modified exponential recovery function depends only on a time constant, τ. The modification results from the fact that a simple $(1 - e^{\frac{-y}{\tau}})$ function in a finite window would introduce a step discontinuity and a slope discontinuity at the end of the window. To alleviate this we simply add a constant and a linear term to form the function $f^R(y)$, where y spans the range $[0, 1]$ over which we scale the recovery range, with the constraints $f^R(1) = 0$, and $\frac{df^R}{dy}(1) = 0$. To fully define the function, we choose in addition that $f^R(0) = 1$ and find the general form,

$$f^R(y, \tau) = \frac{\tau - \tau e^{\frac{1-y}{\tau}} + 1 - y}{\tau - \tau e^{\frac{1}{\tau}} + 1}$$

We find that a linear combination of a small number of these functions with a range of discrete time constants can adequately approximate a continuum of time constants. The lefthand plot in Fig. 20 shows the functions generated for the recovery model using the default τ-related input parameters and

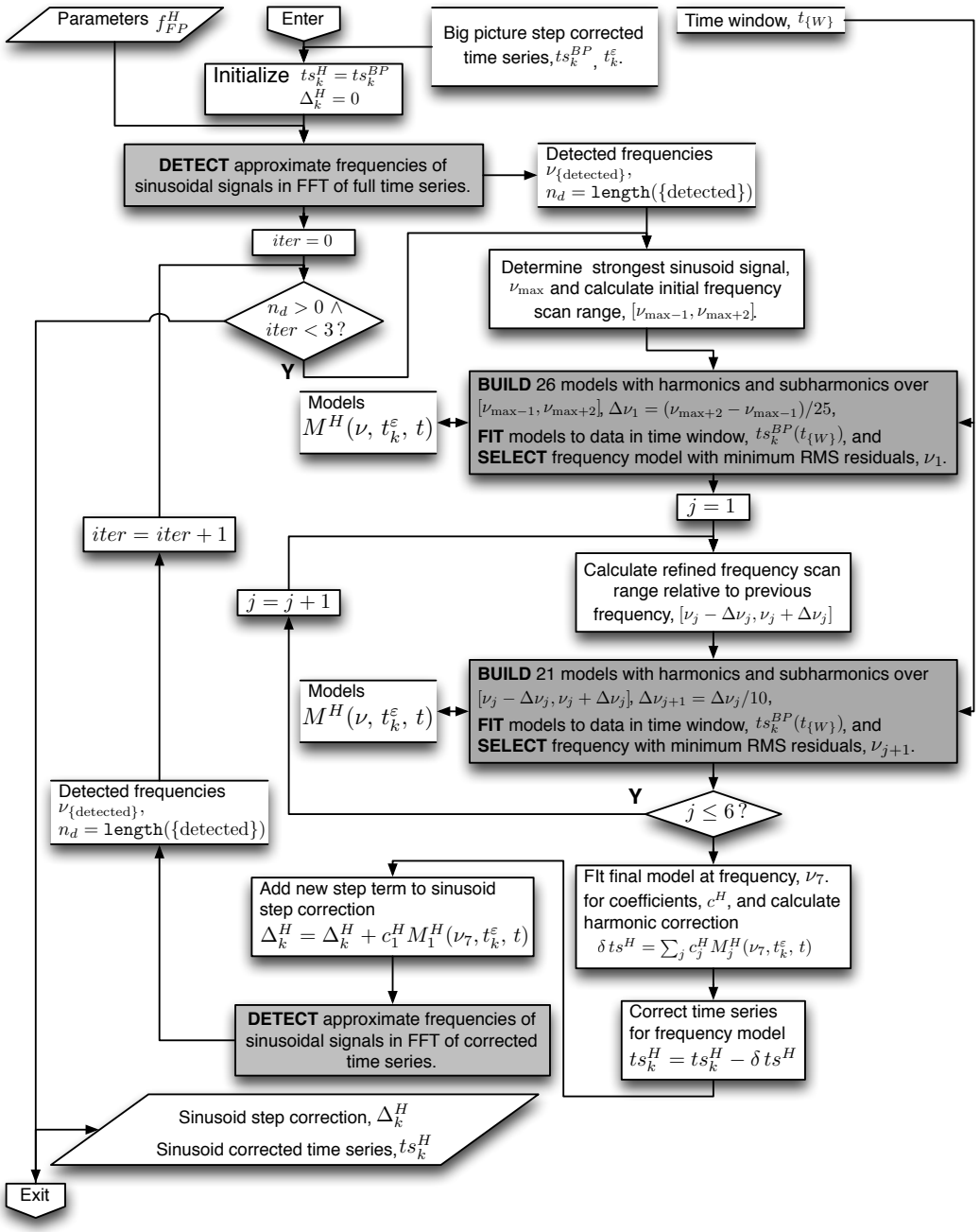


FIGURE 19. **Sinusoid removal segment of SPSD correction algorithm.** This segment requires a big picture-corrected flux time series, ts^{BP}_k along with its SPSD occurrence time, t^{ε}_k, the close-proximity modeling time window, $t_{\{W\}}$, and a parameter defining the recovery window length, $N_{\rm R}$ and polynomial order, $n^{BP}_{\rm p}$. The algorithm iterates through a process of detection modeling and frequency refining to remove sinusoids. The model includes a step, a constant, subharmonics and harmonics of each base frequency sinusoid. The best-fit step component, Δ^H_k and and the model-corrected flux time series, ts^H_k, are output, however only the step component is used in the final overall correction. Appendix C describes the detection steps (light gray blocks) and modeling steps (dark gray blocks). See fig. 15, 38, and 39.

the righthand plot shows a set of examples of the function calculated over a finer range of time constants, along with best-fit approximations to these using the three component basis on the left. These functions perform better in the recovery model than a series of discontinuous polynomials, since they more are more constrained to the expected recover behavior and they do not tend to correlate as much with the stellar component of the data. The use of the linear basis, rather than a nonlinear fit for τ avoids the issues mentioned above regarding nonlinear modeling.

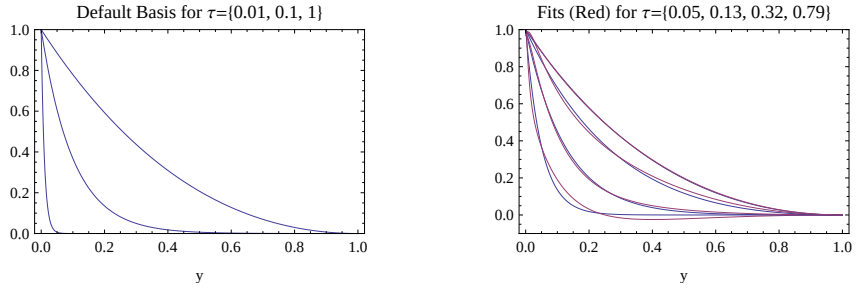


FIGURE 20. **Recovery model basis and completeness.** The figure on the left shows the 3 basis functions produced by the default input parameters which generate $\tau = \{0.01, 0.1, 1\}$. In the figure on the right we simply fit this basis to the same function with a range of time constants to indicating that the 3-component basis is sufficient to represent a continuum of time constants. The blue curves show the original model with indicated τ values, and the red curves are the fits. The agreement indicates that the basis is reasonably complete for representing the modeled effect of the SPSD in the recovery window.

The recovery model is applied to the close-proximity window $\{W\}$ for 2 cases. The first case uses the model including the step function and the second case excludes the step function. We select the case where the standard deviation of the detrended sum of the fit polynomial terms is smaller. We do this to avoid cases where the polynomial components cause the incremental step estimate to be biased. If the selected case is the second case then the final sustained step is only the sum of the components from the first two processing stages, and the final recovery correction is that obtained from the stepless model, otherwise the step-inclusive model results are used in the final output.

Fig. 21 is a flow diagram of the recovery modeling segment of the algorithm. We first model the sinusoid-corrected flux time series, ts_k^H, with SPSD identified at time, t_k^ε, in a time window, $t_{\{Poly\}}$, where $\{Poly\}$ is the included set of cadence indices and k is a target index in {SPSDs} as before. $\{Poly\}$ spans the same range as the sinusoid set $\{W\}$, except data gaps and the recovery gap, which is defined in the big picture modeling section above. The model, which includes just polynomials, is fit using the function `polydeg` for the singular purpose of determining the AIC-derived optimal polynomial order, n_p^R. This is the final ingredient required to enable construction of the recovery model over the close proximity range $t_{\{W\}}$.

The linear model includes $n_p^H + n_\tau + m + 5$ terms, where n_p^{BP} is the input parameter specifying the big picture polynomial order, n_τ is the number of modified exponential terms, and m is a count of CBV provided by coarse PDC-MAP. n_τ is determined by the τ-related input parameters as the length of the set of time constants,

$$\{\tau_1, \tau_2, \ldots \tau_{n_\tau}\} = 10^{\{\log_{10}\tau_{\min}, \log_{10}\tau_{\min} + \Delta\log_{10}\tau, \ldots \log_{10}\tau_{\max}\}}$$

The 5 additional terms are a step (= 0 before 1 cadence prior to t_k^ε, and = 1 during and after 1 cadence prior to t_k^ε) a constant (= 1 for all included cadences), and 3 δ-function terms (= 1 for {1 cadence prior

to t_k^ε, t_k^ε, 1 cadence after $t_k^\varepsilon\}$ and $= 0$ elsewhere). The polynomial terms are represented by the Legendre polynomials $P_j(x)$ for j ranging from 1 to n_p^{BP}. Here x, which ranges from -1 to 1, is a scaled time variable determined from the cadence index, i, by $x_i = 2\frac{i-1}{L_w-1} - 1$, where i is an index in $\{W\}$ which will range between 1 and L_w. The modified exponential terms are represented by the functions $f_j^R(y) = f^R(y, \tau_j)$ for j ranging from 1 to n_τ. Here y, which ranges from 0 to 1, is a scaled time variable determined from the cadence index, i, by $y_i = \frac{i-C_k^\varepsilon-1}{L_R-1}$, where i is an index in $\{W\}$ in the range $[C_k^\varepsilon + 2, C_k^\varepsilon + N_\rho]$, where C_k^ε is the index in $\{W\}$ corresponding to t_k^ε, and N_ρ is the length of the recovery window after the SPSD as defined in the big picture modeling section.

Least-squares fits are performed for this model M and a similar model with the step term excluded M', resulting in sets of coefficients c_J and c'_J with a J corresponding to each term. We decide between M and M', by calculating the polynomial component of each, e.g. $\sum_j c_j M_j, j \in \{\text{polynomial terms}\}$, detrending these using the built-in MATLAB function `detrend`, and then calculating the standard deviations σ, and σ'. The selected model is the one with smaller variation in its polynomial component. If M is the selected model, then the J corresponding to the step term ($J = 1$ in fig. 21) identifies the coefficient which gives a final adjustment of the sustained step height, yielding the result,

$$\Delta_k^S = \Delta_k^{BP} + \Delta_k^H + c_k^1\,\Theta(t, t_k^\varepsilon - \delta_1),$$

otherwise the sustained step correction is just $\Delta_k^S = \Delta_k^{BP} + \Delta_k^H$. In this case we define

$$\Theta(t, t') = \left\{ \begin{array}{cccc} 0 & t & < & t' \\ 1 & t & \geq & t' \end{array} \right\}.$$

The recovery correction is

$$\Delta_k^R = \sum_j c_j^x M_j^x,\ j \in \{\delta-\text{function terms}\} \cup \{\text{exponential terms}\}$$

where $M^x = M$ and $c^x = c$ if M is selected, and $M^x = M'$ and $c^x = c'$ if M' is selected. The resulting SPSD corrected time series is $ts_k^C = ts_k - \Delta_k^R - \Delta_k^S$.

3. Performance

3.1. **Detection Filter Performance.**

Comparison with alternatives. We apply the multi-scale SPSD filter described above, and several others to the sample SPSD-containing time series shown in fig. 1, and compare the signal-to-noise obtained in the various cases. For the comparison, we used the filter coefficients derived from the Savitzky-Golay + discontinuity models described in §2.1 with the default long-window and short-window parameters from Table 1. These are shown as the left-most plots in the top 2 rows in fig. 22. We also compare a pair of simple examples inspired by the use of gaussian derivatives in optimal edge detection by Canny.[5] For these we use curves calculated from first derivatives of Gaussians. In one case we use a single gaussian width parameter, which is 1/3 of our 96-cadence long-window half-width. For the other case, in addition to the $\sigma = 32$ term, we combine first derivatives of Gaussian curves with widths of 16, 8, 4, and 2 cadences by simple sum. The resulting filter kernels are shown in the third and fourth rows in fig. 22.

The second column of plots in fig. 22 shows the result of applying each filter to the time series shown in fig. 1. The third column, labeled "step response function" is the local response, as nonzero filter coefficients interact with the step in that same light curve. As such, it is not the ideal step response, but an illustrative response to a step + noise + a small exponential recovery. For quantitative comparison, we

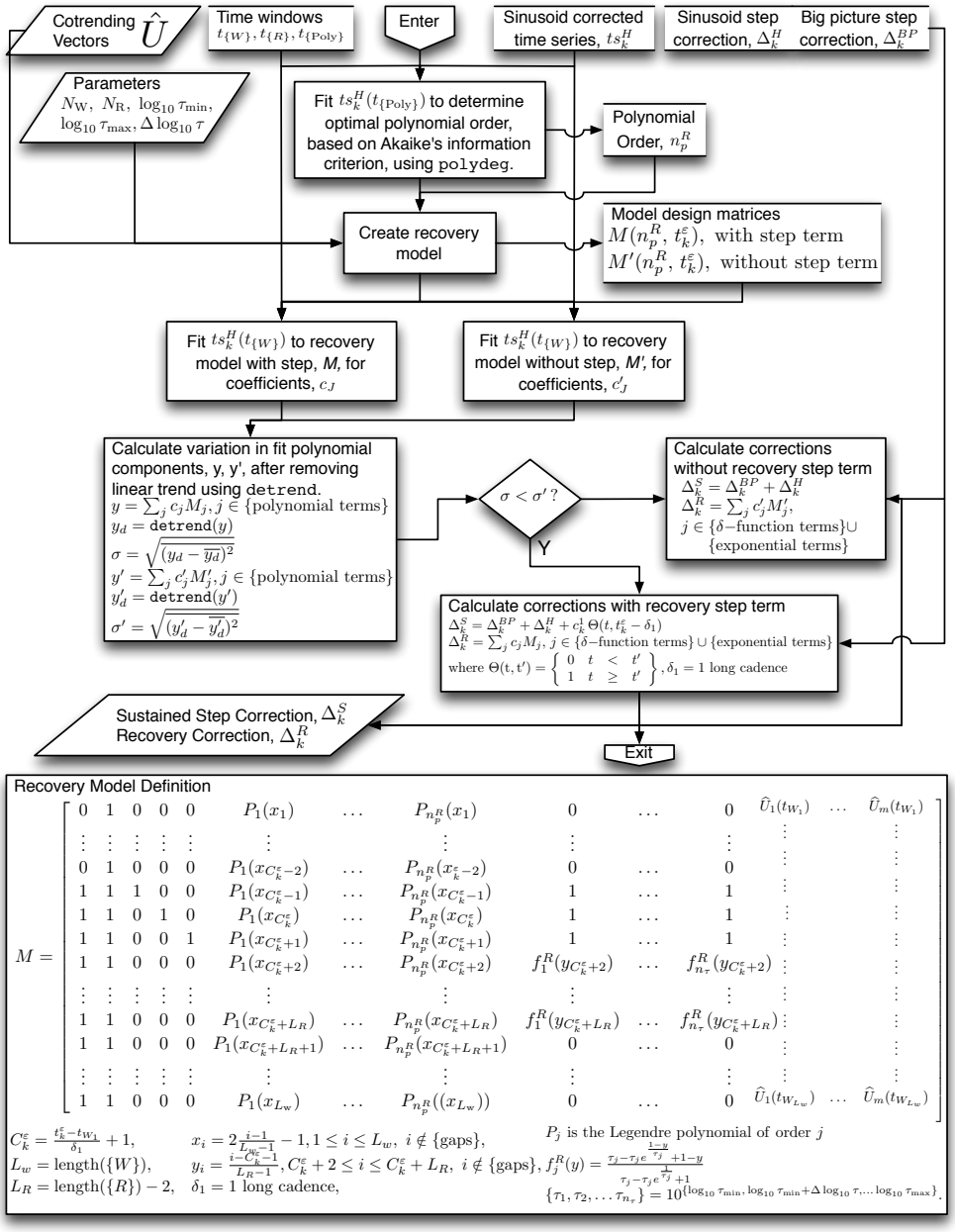


FIGURE 21. **Recovery modeling segment of SPSD correction algorithm..** This segment requires a sinusoid-corrected flux time series, ts_k^H along with its SPSD occurrence time, t_k^ε, , the coarse MAP CBVs, $\widehat{U}$, three modeling time windows, $t_{\{W\}}$, $t_{\{R\}}$, and $t_{\{\mathrm{Poly}\}}$, the parameters defining window lengths, N_{W} and N_{R}, and three parameters defining the sequence of exponential time constants, τ, $\{\log_{10}\tau_{\min},\ \log_{10}\tau_{\max}, \Delta\log_{10}\tau\}$. The first step in the algorithm determines an optimal polynomial order. It then fits a model with and without a step to evaluate whether including an additional term in the final step correction is necessary. The model includes a step, a constant, polynomial components, exponential recovery components, three recovery δ-functions, and the CBVs, but all except the step and recovery terms are nuisance parameters. The sustained step correction Δ_k^S and and the recovery correction, Δ_k^R, are the only outputs. See fig. 15.

list the signal-to-noise ratios (SNR), where the signal is the measured response at the step location and the noise is the standard deviation of selected other filter output. The fourth column shows the SNR results when we select only filter output more than ±96 cadences from the step for the noise estimate, i.e. where the response is not affected by the step. For the right-most column, we include all the data except the step cadence in the noise calculation.

We selected the multi-scale SPSD filter because:

- for most light curves it maintains a SNR comparable to a longer time-scale filter, making it sensitive to the smaller SPSDs, and therefore a larger quantity,
- it produces a highly localized response, improving performance near potentially confounding features and increasing transit perception for vetoing purposes.
- it accounts for almost all red noise sources in the modeled components, and
- it is efficient to implement requiring only a single pass to generate detection statistics across a light curve.

In comparison, the short S-G filter is less effective at reducing noise, while the long SG-filter, although having the best SNR far from the step, has a complex step response function. This complexity would limit our ability to find SPSDs near long gaps affected by other systematics, or lead to difficult-to-interpret results for transits and flares because of the interference between responses to upward and downward going features. The lack of localization of the Gaussian derivatives results in similar limitations. In addition, the fact that the coefficients of these all have the same sign on a given side of the center implies they respond to nonzero light curve slopes, which are a common systematic as well as astrophysical features, thus to implement these kernels would require additional steps of high-pass filtering, and there is no systematic means of selecting scales to add for the multi-scale case. Algorithms implementing automatic scale selection have been developed [6] for 2D image processing, but again would introduce more complexity, especially in dealing with the discrimination among features of systematic or astrophysical origin and SPSDs. The derivative of gaussian is optimal for white noise. We have substantial red noise, trends and features in the data, requiring whitening before application of such a filter, and whitening itself is problematic in the presence of SPSDs. That is partly why they need to be removed before later whitening steps in the pipeline. The multi-scale SPSD filter kernel is robust against red noise sources in the data.

Period Response. At the start of development of this algorithm, we planned that sinusoids would be removed in advance of SPSD-detection. Therefore, the filter was not designed to find SPSDs comparable or smaller than the amplitude of a high frequency signal, it is essentially a high pass filter. Later the sinusoid removal step was moved after SPSD detection, leading to reduced SPSD sensitivity for targets containing a large fraction of power in the high frequency portion of the power spectral density distribution(PSD). Fortunately, very few targets exhibit this characteristic and so the reduction in SPSD detection efficiency is very small. In this section we discuss the period response of the multi-scale SPSD filter.

The uppermost plot in fig. 23 shows the period response and indicates that a filter cutoff of 70 long cadences or ~ 1.5 days. A sinusoid with amplitude 1 has standard deviation, and therefore an equivalent noise contribution of $1/sqrt2$. Taking this into account, the three other plots in fig. 23 indicate the noise contribution in the filter output of a sinusoidal input of unit amplitude. The second plot covers the range out to > 2 month periods, after which the filter response is < 140 db. The third plot covers the period range of the nominal filter length, 4 days, and the bottom plot shows the rather complex response to frequencies above the Nyquist frequency. Any target exhibiting strong variation in a band with period < 70 long cadences is likely to suffer diminished sensitivity to SPSDs. The degree of reduction is plotted

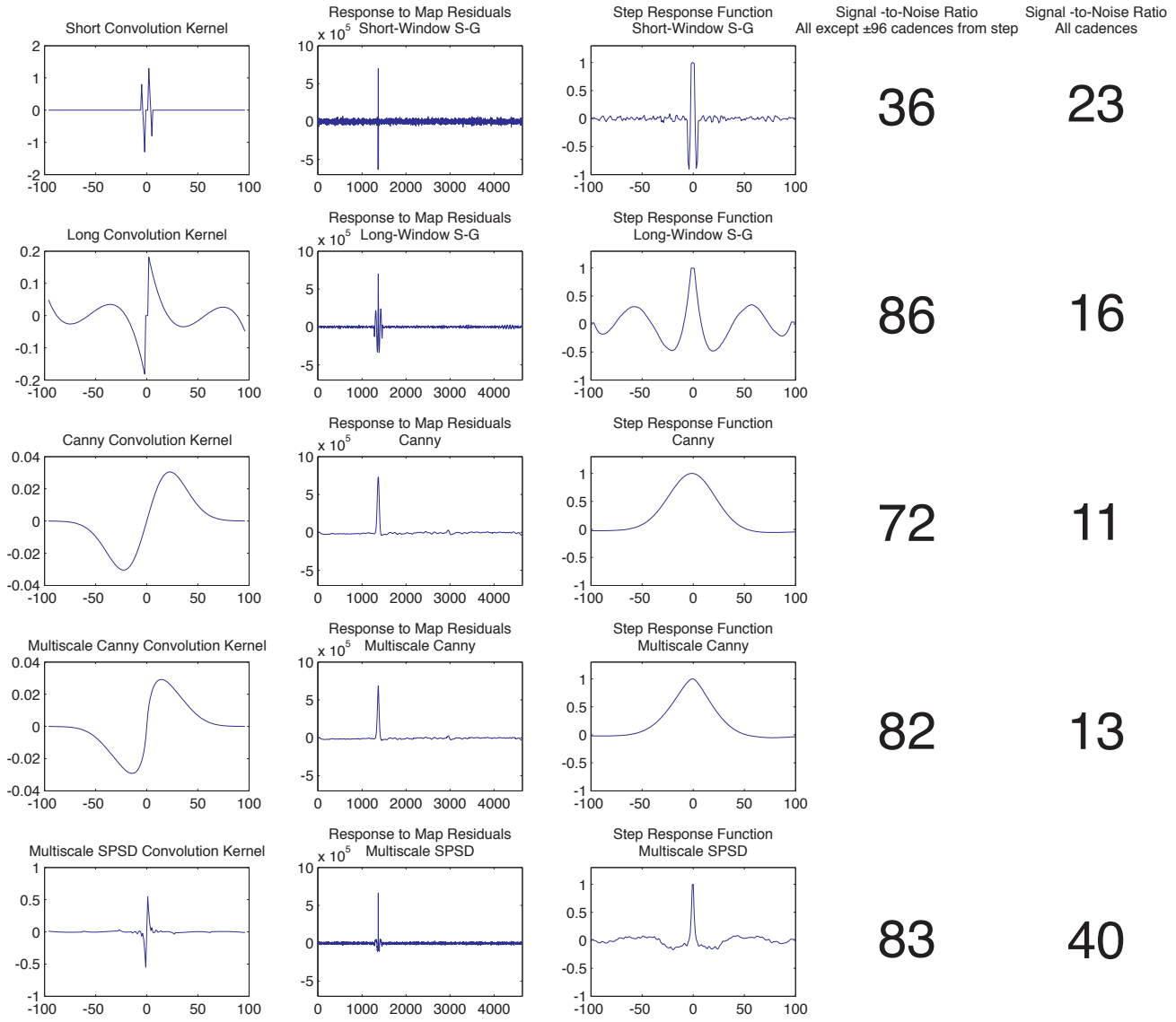


FIGURE 22. **Comparison of a sample of detector filters.** The multiscale SPSD filter used in this algorithm is shown at the bottom and compared with 4 others as labeled. The filter is applied to the time series shown in fig. 1. The signal to noise ratio columns indicate the noise response and localization. Both compare favorably with the other filters which act on similar time scales. The high frequency variations in this target are far above the shot noise limit, but the large size of the SPSD makes it a good illustration.

as a function of the size of the sinusoidal signal above noise in fig. 24. The value 1 indicates no drop and 0 indicates no sensitivity in this plot.

A survey of targets indicates that that only 1-2% have significant signal above noise in the transmitted band, so this is the reduction in detection efficiency. On average this amounts to < 1 missed SPSD per

channel per quarter. While this suboptimal feature of the detection algorithm could be improved, there are currently no plans to do so.

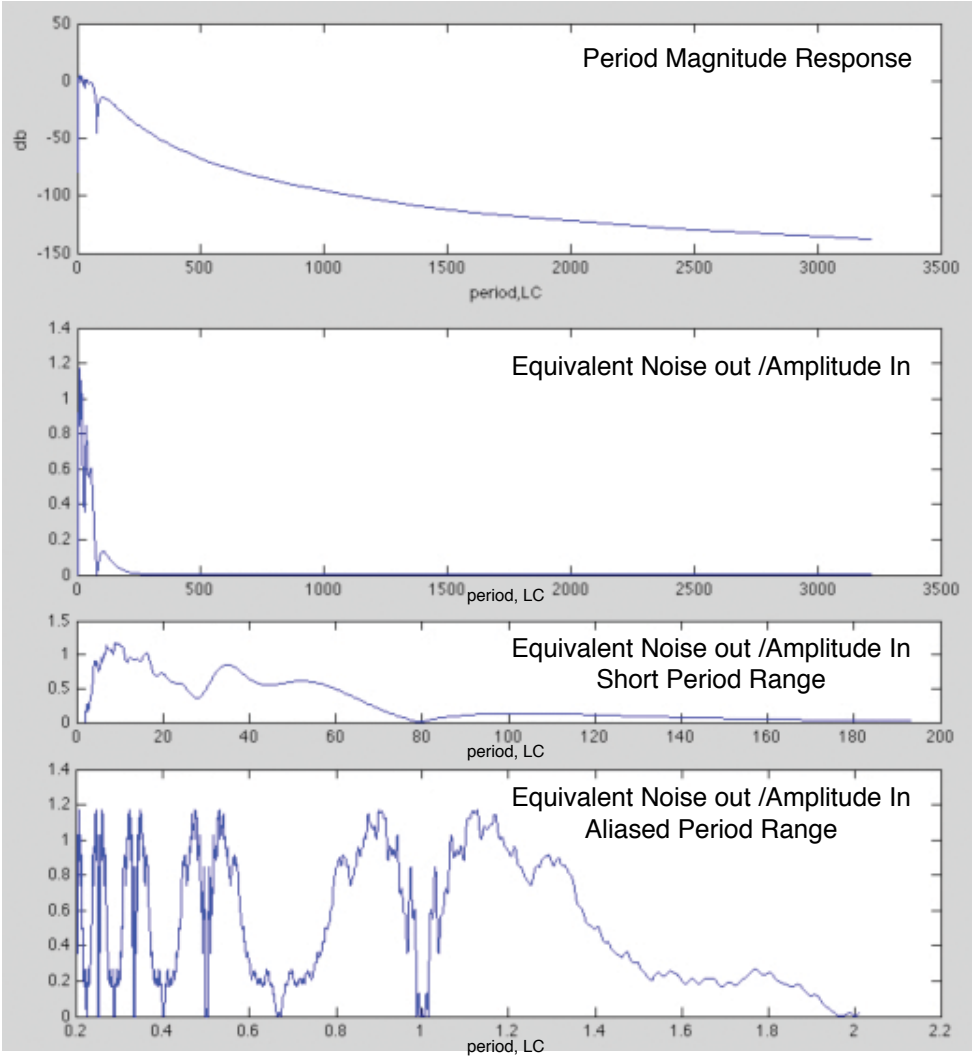


FIGURE 23. **Period response of the multiscale SPSD Filter.** Signals at high frequencies, $> 0.7\,\mathrm{days}^{-1}$ pass through the filter, acting as a noise source. The response is a complicated function above the Nyquist frequency.

3.2. **Detection Algorithm Performance.** The three plots in figure 25 summarize the results of the detection algorithm for all targets on Channel 7.3 during Q5. The upper left plot is simply the maximum detection statistic, ε_k (labeled MxDS), as a function of the target index, k. It indicates a noise floor, predicted by extreme value theory as described in Appendix B, with numerous SPSDs rising above it. The lower left plot shows the distribution of these in logarithmic bins. The peak is $\sim 4\sigma$, which is the expected value, and the gaussian distribution around the peak, are the result of the filter response to noise. A tail is evident in the distribution above 6σ representing SPSD candidates, and this extends down into the noise.

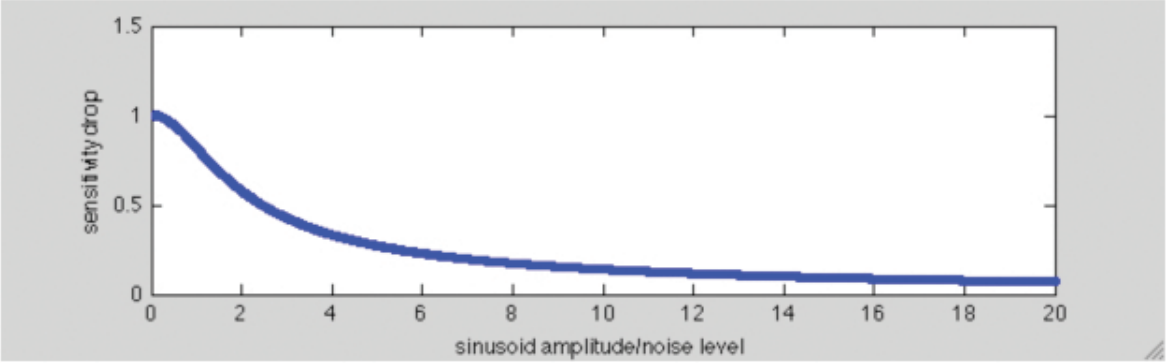


FIGURE 24. **Effect of high frequency sinusoids on SPSD sensitivity.** The plot shows the fraction of full sensitivity as a function of the ratio of an in-band sinusoid amplitude over noise. It approaches an inverse function for sinusoid amplitude/noise level > 2.

The scatter plot on on the right shows ε_k vs. the sum, Σ_k, of ε_k and the the minimum in a ± 96 cadence window around the maximum. The blue points are non-candidates, the red points are candidates that were rejected in validation, and the green points are confirmed SPSDs. Many blue points which have large ε_k do not become candidates because they are accompanied by a nearby minimum which substantially reduces the sum, Σ_k. The validation primarily discriminates among candidate near the noise floor, as indicated by the locations of the red points. As expected, the confirmed SPSDs cluster around the line $\varepsilon_k = \Sigma_k$.

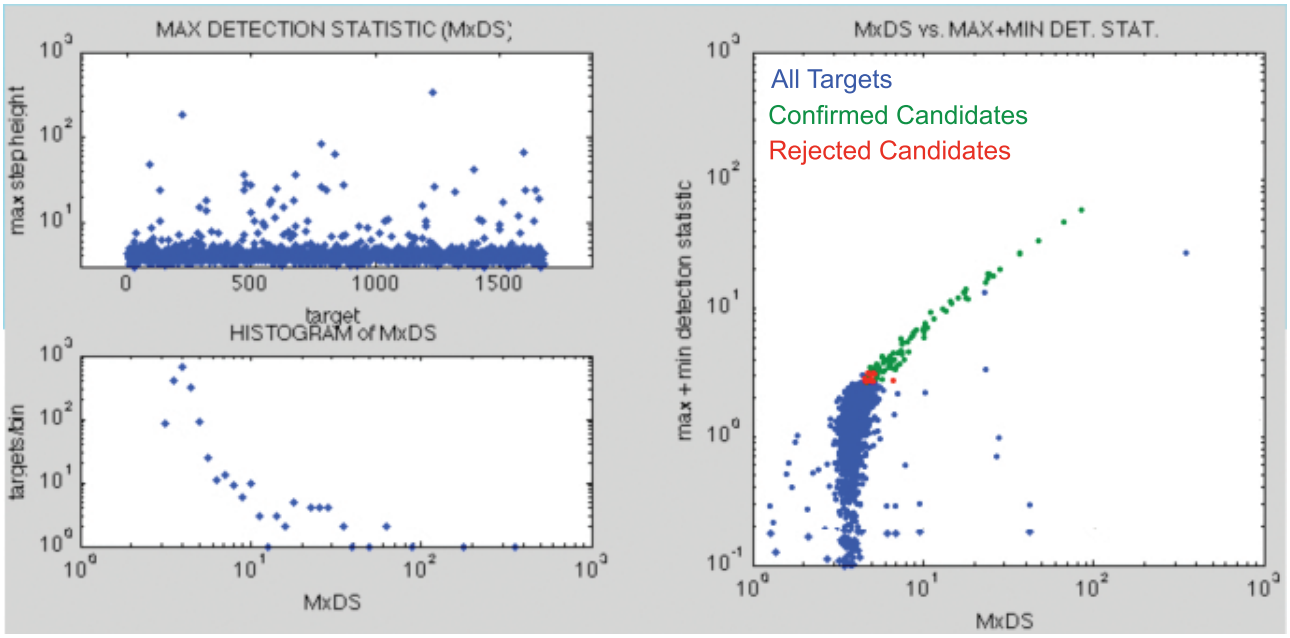


FIGURE 25. **Maximum Detection Statistic Distribution.** The upper left plot shows the maximum detection statistic, ε_k, vs. target index, k. The lower left is a histogram showing the distribution of ε_k. The scatter plot on the right shows ε_k(max) vs. the sum, Σ_k (max+min), with colors representing the candidate and confirmed candidate discrimination.

Performance evaluation on data containing real SPSD signatures requires access to ground truth. Since manual identification of all SPSDs in an entire channel is extremely costly and somewhat unreliable we supplemented limited manual evaluation (see above) with simulations. Simulated data sets were created by injecting artificial SPSDs into a "clean" channel of Q7 flight data from which all targets containing visually identifiable SPSD signatures had been removed. Six such data sets were created by randomly injecting sets of 100 SPSDs into the time series of 12th magnitude targets ($11.5 \leq m \leq 12.5$). Simulated sensitivity drops ranged from 0.1% to 2% and were constant within each test case. The fig. 26 shows estimated performance as a function of discontinuity step size in parts per million (ppm). The fall-off occurs as the SPSD distribution extends into the noise distribution, as can be seen from the distributions of hits and misses shown in fig. 27.

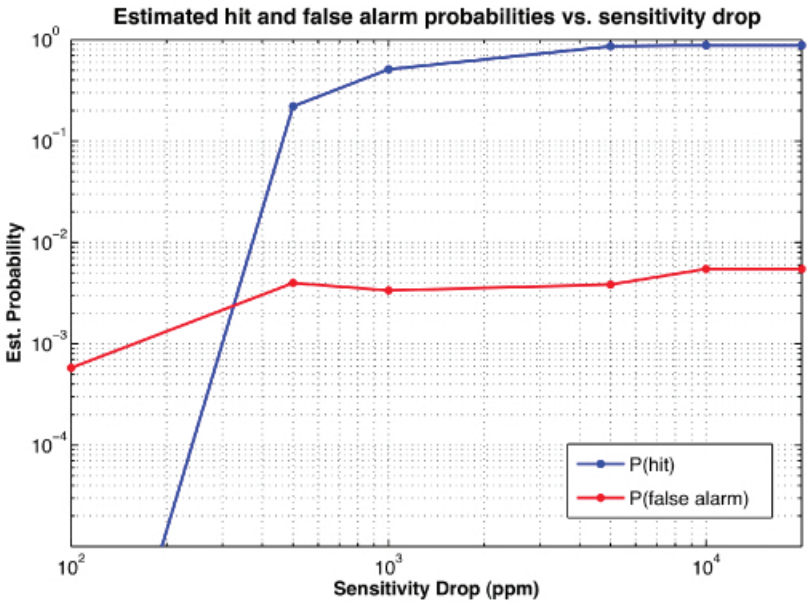


FIGURE 26. **Detection Algorithm Performance.** The estimated performance as a function of discontinuity step size is shown for simulated steps.

A manual survey of maximum detection statistic locations was performed to classify ground truth for module output 13.1 during Q10 using Version 8.1 pipeline code and the default values for $f_{FP} = 0.005$. The results are shown in Table 9. The 2 off diagonal entries represent false positives (lower left) and false negatives (upper right). The observed false positive rate of 0.007 is in good agreement with our specification, suggesting that the three- step standardization process is successfully producing a standard noise distribution, $\sim N(0, 1)$. We cannot predict the false negative rate *a priori*. If one weighs a false negative as being equally undesirable as a false positive, then setting f_{FP} to a value where the two rates are measured to be the same is likely to be the the best we can achieve. Therefore, the table suggests that 0.005 is a reasonable value to use.

3.3. **Correction Algorithm Performance.** Correction performance is difficult to quantify using real SPSDs because of the often complex light curves. We perform visual inspection on large numbers of corrected time series to search for potential flaws in the algorithm. To illustrate we include some examples. Figure 32 contains 3 examples from Q5 from prototype runs on module outputs 7.3 and 13.4. Figure 31 is

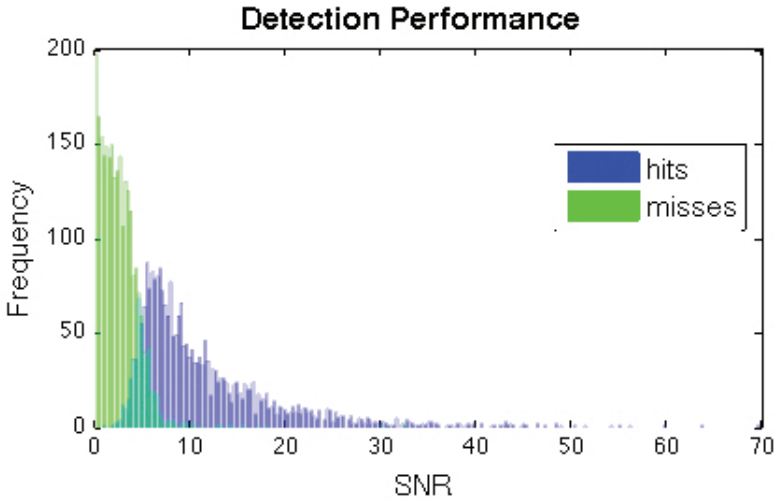


FIGURE 27. **Detection performance against simulated ground truth.** Simulation testing allowed for quantification of detection performance relative to ground truth. The distributions illustrated are consistent with our expectations that SPSDs begin to blend into noise at the $\sim 4.5\sigma$ level. This figure is an output of pipeline ver. 8.1testing on Q7 data.

	SPSD detected	*SPSD* not detected
SPSD	47	15
no *SPSD*	14	1913

TABLE 9. **Detection Algorithm Confusion Table.** The results of a manual tally of SPSDs are tabulated with the observed "ground truth" in rows and detector results in columns. This is derived from Q10 data from channel 13.1.

an example from a Q7 pipeline run. For a more quantitative ground truth evaluation we used the injected simulated SPSDs described in § Section:detectPerf.

Some example injected SPSDs are shown in fig. 28 for a sampling of signal-to-noise ratios. The correction algorithm satisfactorily corrects the light curves as shown in the ground truth example in fig. 29. To evaluate the correction over the injected set we calculate the RMS error introduced by the injected SPSD, e.i. the RMS of the injected SPSD signal itself for each target, $RMSE_{\text{injected}}$. After correction we repeat the calculation to get $RMSE_{\text{corrected}}$. Fig. 30 shows the distribution of the V&V performance metric, percent RMSE reduction, which is $(RMSE_{\text{injected}} - RMSE_{\text{corrected}})/RMSE_{\text{injected}} \times 100$. The correction algorithm is quite effective with $< 2\%$ of the light curves showing a small percentage improvement.

The plot also shows correction performance for targets with false alarms, but this turns out to be a bit counterintuitive. We would expect a well-populated bin at zero, since targets containing only false alarms have an undefined RMSE reduction which gets set to zero by the analysis code. However, there are few counts in the zero bin of the histogram. This is because false alarms seem to occur mostly in targets that have already been corrected for an SPSD. For the simulated case, there appears to be an increase the likelihood of a false alarm on successive iterations, however the corrections in these cases are minimal and benign, as indicated by the fact that for most targets containing false alarms, there is a significant improvement in the RMSE relative to ground truth.

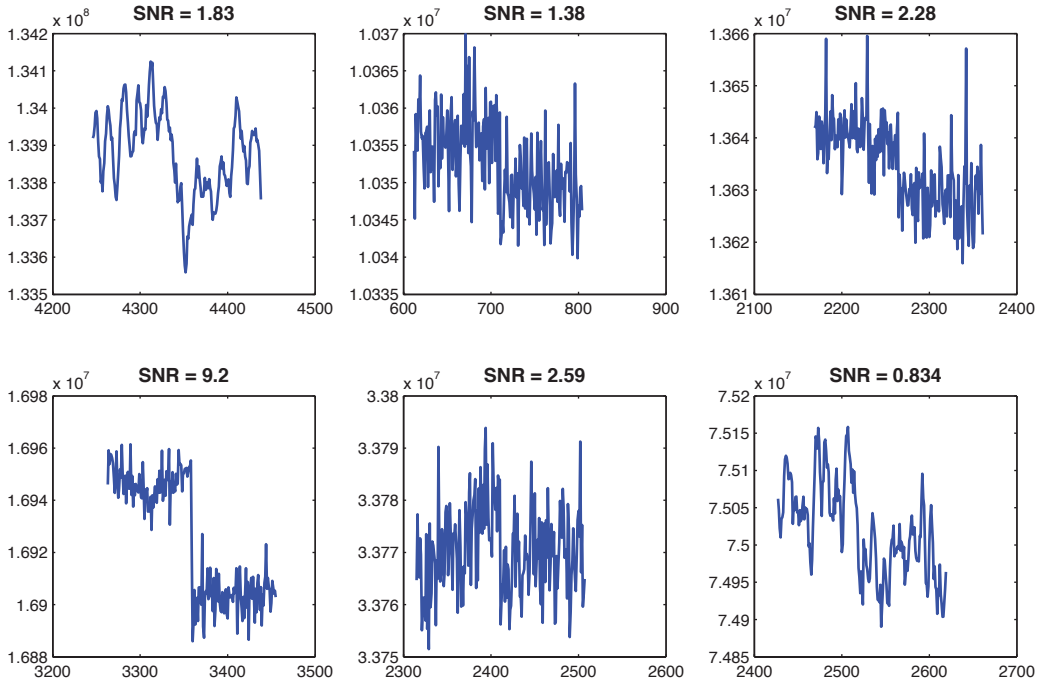


FIGURE 28. **Example simulated SPSDs ground truth.** Simulations required a process of removal of SPSDs from a sample of light curves followed by injection of simulated SPSDs to establish a ground truth. The figure illustrates some example injected signals at various SNR levels.

REFERENCES

[1] Jenkins, J. M., Caldwell, D. A., Chandrasekaran, H., et al. 2010, ApJ, 713, L87
[2] Smith, J.C. et al. 2012. submitted to *PASP*.
[3] Stumpe, M.C. et al. 2012. submitted to *PASP*.
[4] Savitzky, A. and Golay, M. 1964. Analytical Chem., 36 (8), 1627-1639
[5] Canny J. 1986, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8, 679-714.
[6] Lindeberg, T. 1998 *International Journal of Computer Vision*, 30 (2), 117-154.
[7] Castillo, E. 1988. *Extreme value theory in engineering.* Academic Press, Inc. New York.
[8] Gumbel, Emil J. 1958, *Statistics of Extremes.* Columbia University Press, ISBN 0-483-43604-7
[9] Pickands, J. 1975, *Annals of Statistics*, 3, 119131.
[10] `www.mathworks.com/matlabcentral/fileexchange/29865-optimal-polynomial-fitting/content/polydeg.m`

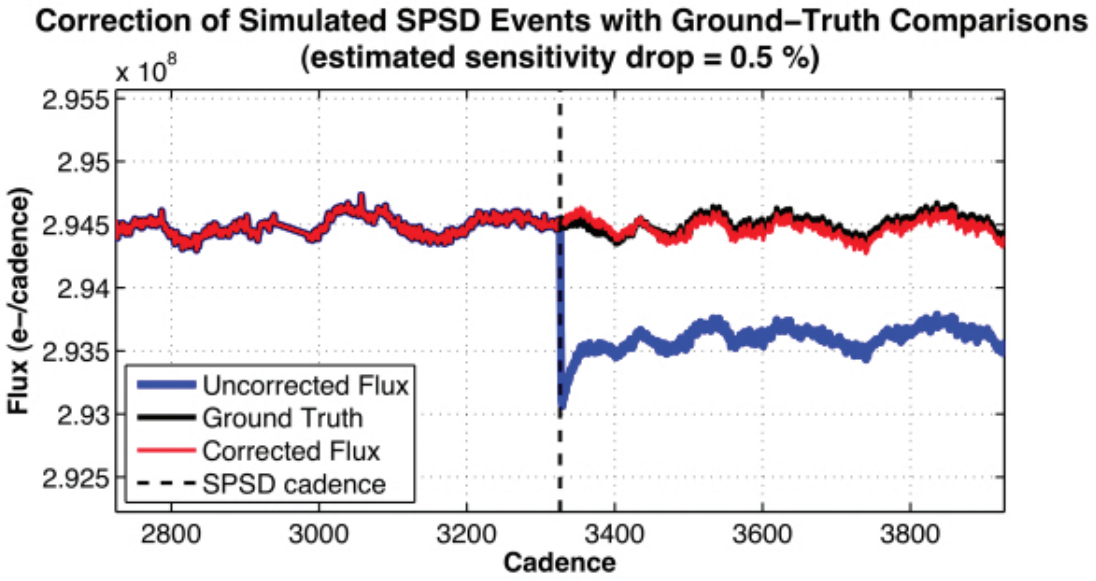


FIGURE 29. **SPSD ground truth correction example.** Simulation testing allowed for quantification of correction performance relative to ground truth. The figure illustrates the typically high-quality reconstruction of the original signal.

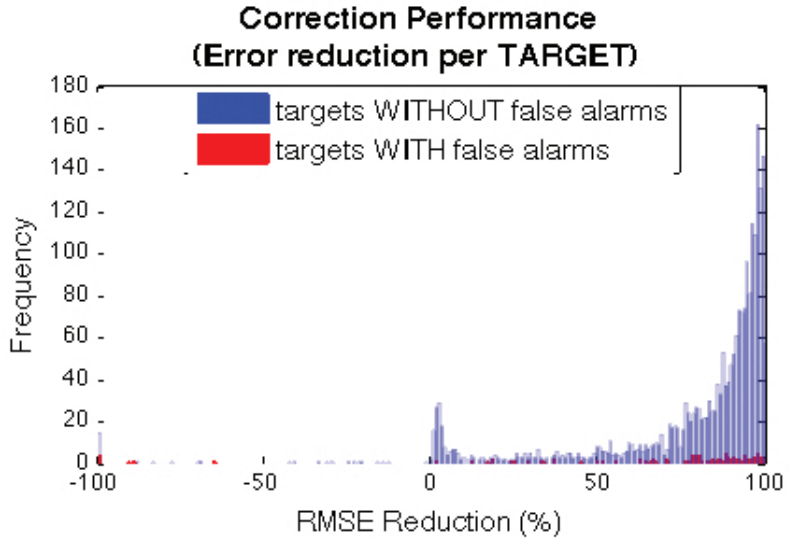


FIGURE 30. **SPSD ground truth correction summary.** The RMS error introduced by a simulated SPSD is known. After correction the RMS error is recalculated and a percent reduction for each target is derived. The distribution indicates that the correction is effective for over 98% of targets. Despite the distribution of false positives, the corrections associated with these are quite benign.

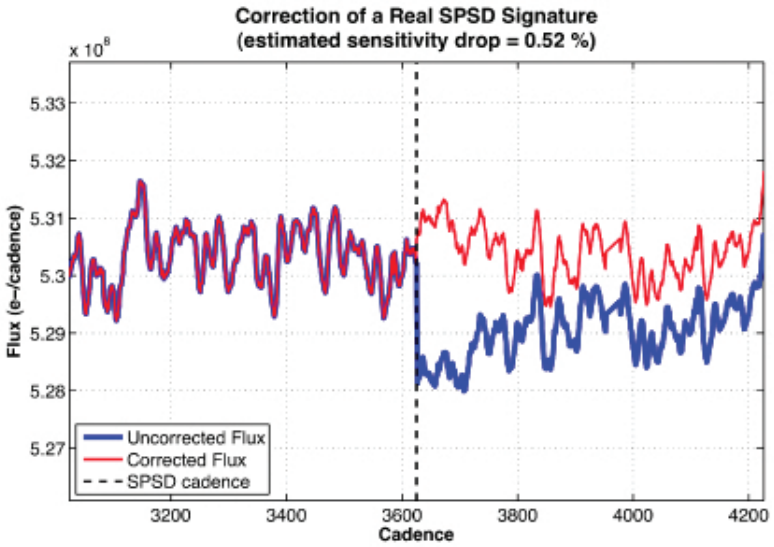


FIGURE 31. **SPSD correction examples from Q7.** 1984 light curves from a single Q7 channel were manually inspecting and the results compared with the SPSD detector's output. The figure shows an example Q7 correction.

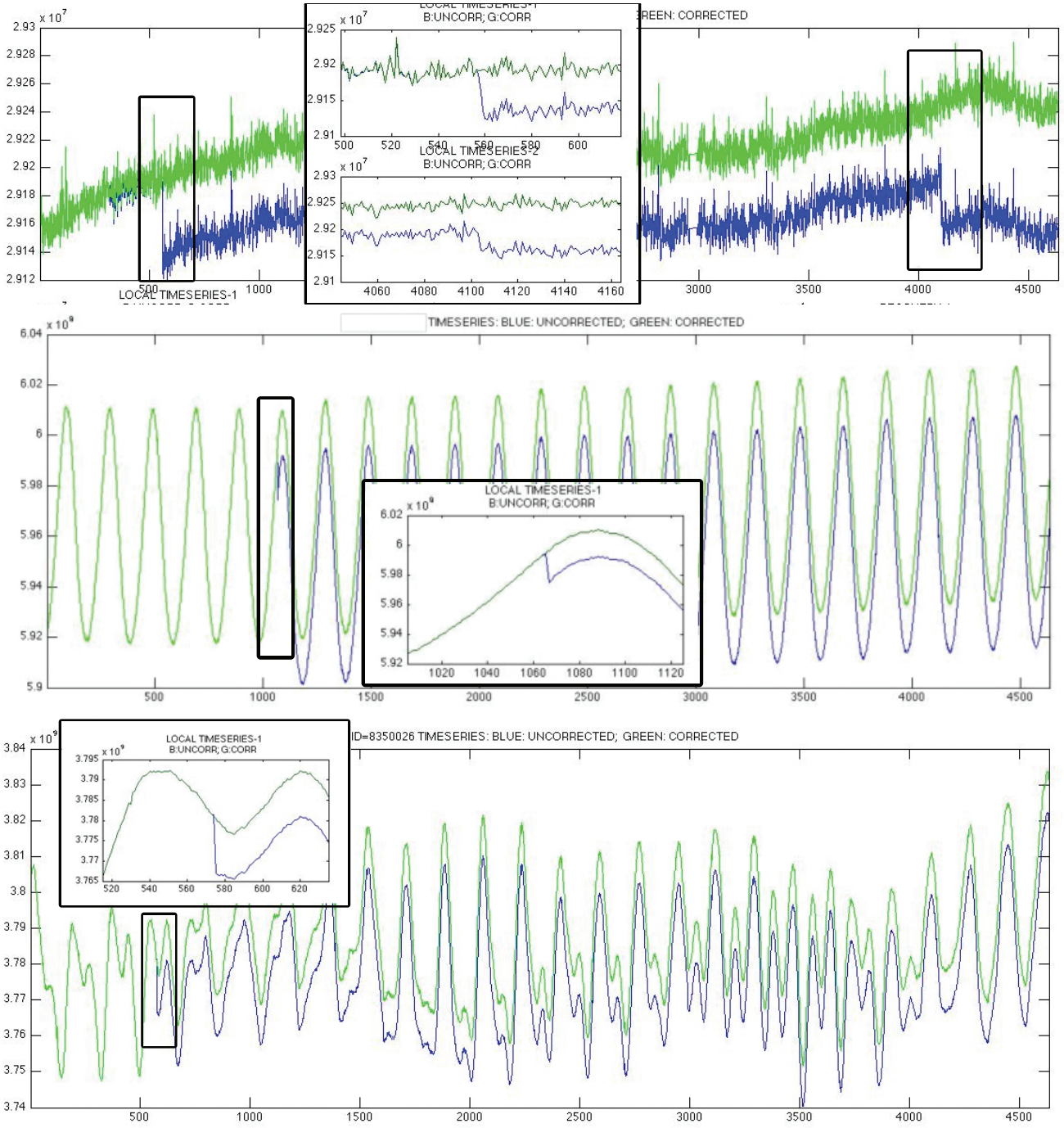


FIGURE 32. **SPSD correction examples from Q5.** Examples of algorithm performance on 3 Q5 flux time series. The top shows a time series with 2 SPSDs, the middle example has a clean sinusoidal signal, and the bottom has a more irregular time series. In each case the corrections appear reasonable.

Appendix A. Details of Preconditioning

The flux time series, F, input the SPSD detection algorithm, undergo a series of preconditioning steps summarized in fig. 12, to produce a preconditioned time series, P, padded by length $\lfloor N_{\mathrm{LongWindow}}/2 \rfloor$ on each end. This processing permits effective filtering near ends and gaps where sudden signal changes could otherwise result is spurious output. The following figs. 33, 34, 35, 36 provide details regarding various segments in this algorithm. Fig. 33 and caption describe single cadence gap filling.

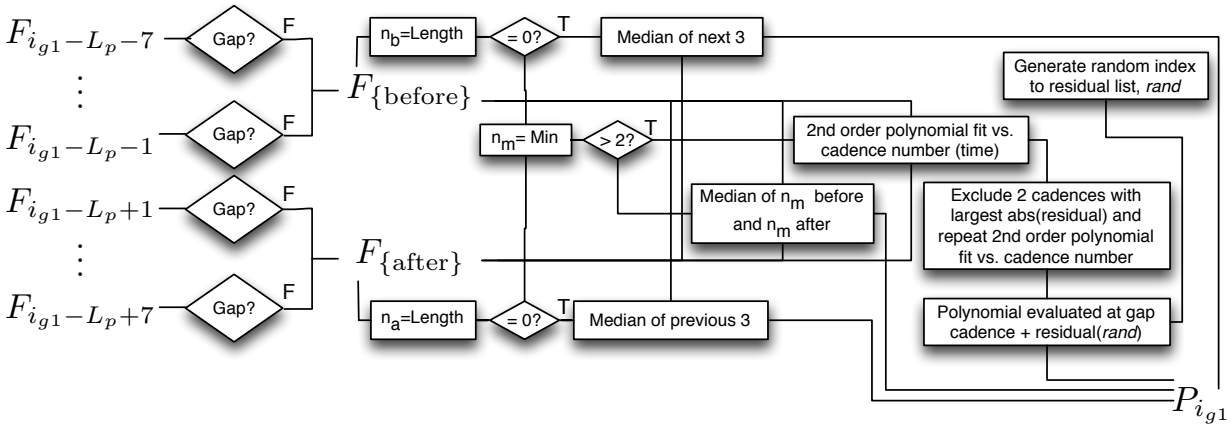


Figure 33. **Single-cadence gap filling for preconditioning segment.** The algorithm determines the fill-value for a single cadence gap, $P_{i_{g1}}$, using the input time series data, F, in a window at most ± 7 cadences around the gap. We choose i_{g1} as the index to the location in the padded time series, so the corresponding input time series location is $i_{g1} - L_p$, where L_p is the single-side padding length in cadences. These data, excluding other gaps, are fit to a second order polynomial. The algorithm then locates and excludes the two largest residuals, and repeats the fit. $P_{i_{g1}}$ is the fit value at the cadence plus a randomly selected residual value which prevents the noise level from dropping at each single cadence gap. The exceptional cases are: if the gap is at the beginning or end of the time series, then $P_{i_{g1}}$ is the median of the 3 nearest valid data points; and if either side of the gap has ≤ 2 valid points, then $P_{i_{g1}}$ is the median of an equal quantity of values before and after the gap. See fig. 12.

Fig. 34 describes the filling of multi-cadence gaps.
Fig. 35 describes the process for padded the ends of the time series.
Fig. 36 describes the 3σ outlier detection and removal process.

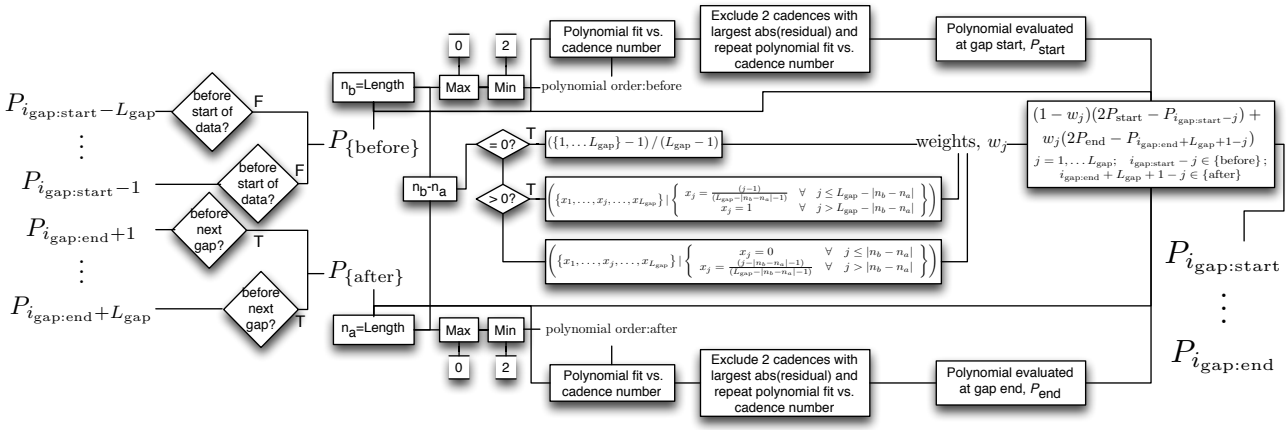


FIGURE 34. **Multi-cadence gap filling for preconditioning segment.** The algorithm determines the fill-values for multi-cadence gaps, of length, L_{gap} cadences $\{P_{i_{\text{gap:start}}},\ldots P_{i_{\text{gap:end}}}\}$, using valid or single-cadence-filled data in a window at most L_{gap} cadences around the gap. The data before and after are fit to second order polynomials separately with 2 point exclusion as in the single-cadence case to extrapolate estimates for the starting and ending fill-values, P_{start} and P_{end}. We apply a weighting function, w, ranging linearly from 1 at the start to 0 at the end of the gap, to time-reversed pre-gap data with offset to begin at value, P_{start}. These are added to a similar term derived from time-reversed post-gap data with weighting function $1-w$ and offset to end at value, P_{end}, to produce smooth fill-values with consistent noise levels. The exceptional cases are: if the gap is nearer than L_{gap} to either another multi-cadence gap, or the beginning or end of the time series, and this leads to a different quantity of pre- and post-gap data points, then w is adjusted as indicated to account for the difference; and if either side of the gap has ≤ 2 valid points, then the polynomial order and number of exclusion points is reduced to avoid rank deficiency in the fit. See fig. 12.

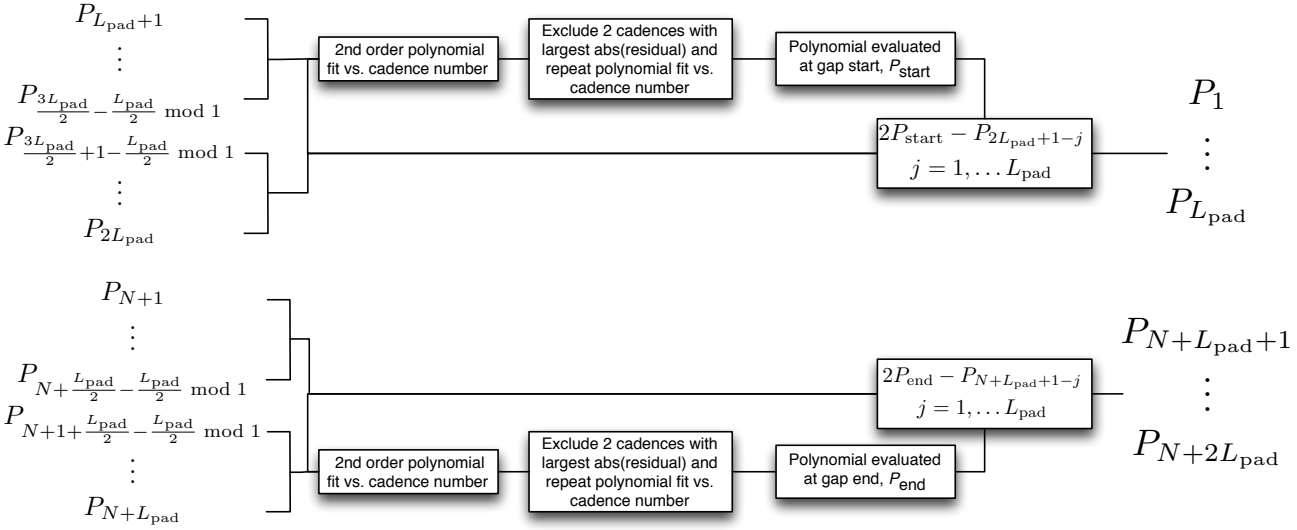


FIGURE 35. **End-pad generation for preconditioning segment.** The algorithm generates values for padding the initially N-cadence time series start and end, of length, $L_{\rm pad}$ cadences, $\{P_1, \ldots P_{L_{\rm pad}}\}$, and $\{P_{N+L_{\rm pad}+1}, \ldots P_{N+2L_{\rm pad}}\}$, using valid, single-cadence-filled, or multi-cadence-filled data in a $L_{\rm pad}$-cadence window at the time series start and end, respectively. For each of the start and end cases, the nearest $\lfloor L_{\rm pad}/2 \rfloor$ data are fit to second order polynomials with 2 point exclusion as in the single-cadence case to extrapolate estimates for the initial pad-values, $P_{\rm start}$ and $P_{\rm end}$. The full-$L_{\rm pad}$-cadence time series data is time-reversed and offset to the initial pad-value, to produce smooth pad-values with consistent noise levels. See fig. 12.

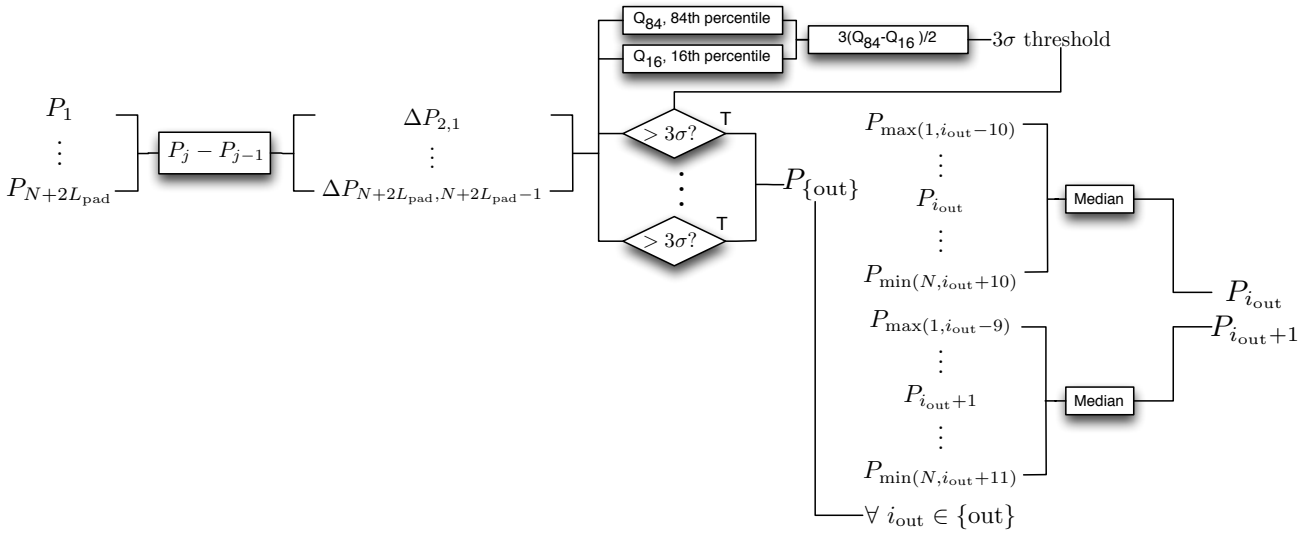


FIGURE 36. **Outlier detection and correction for preconditioning segment.** The algorithm replaces 3σ outliers and values immediately following outliers with corrected values, using valid, single-cadence-filled, multi-cadence-filled, or padded data in a ± 10 cadence window around and including the values to be replaced. It estimates σ from half the difference between the 84th and 16th percentiles of the time series first differences. The correction is a simple median of the window data which preserves the location of any step discontinuity in the time series. See fig. 12.

Appendix B. Details of Detection Threshold Calculation

In the detection algorithm, we need to understand the expected distribution of the detection statistic, which is the thrice-standardized detection filter output, so that we can distinguish among fluctuations due to noise, astrophysically induced flux variations, and SPSDs. In this appendix we derive the thresholds that define these discriminations. Fig. 37 is a flow diagram of the algorithm. This derivation is based on extreme value theory [7] [8] [9].

We assume that the filter output away from actual SPSDs samples a Gaussian noise distribution, and that, as a result of our standardization steps, the detection statistic for all targets samples the standard Normal distribution, $N(0,1)$. Any observation of the standardized filter output, x_{obs} which is not an SPSD is assumed to sample this distribution, $x_{obs} \sim N(0,1)$. The probability density function (PDF) is $f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{x^2}{2}}$, and the cumulative distribution function (CDF) for x_{obs} is $P(x_{\rm obs} \mid x_{\rm obs} \leq x) = \int_{-\infty}^{x} f(x')dx'$.

We next observe that SPSDs occur on average $\ll$once per time series so it is sufficient to test only the maximum value of each time series, to find most SPSDs, then iterate to find any remaining time series with > 1. To evaluate whether the max value is consistent with noise we first calculate the probability that the maximum value in an N_s-sample time series will be $\geq \varepsilon$.

$$P(\max(x_{\rm obs}) \geq \varepsilon \mid N_s \text{ samples}) = 1 - P(x_{\rm obs} \mid x_{\rm obs} \leq \varepsilon)^{N_s}$$

Here $P(\max(x_{\rm obs}) \geq \varepsilon \mid N_s \text{ samples})$ gives the distribution of maxima expected from repeatedly finding the maximum of N_s-long samples from $N(0,1)$. Then, given a desired false positive rate, f_{FP}, the threshold for saying a maximum is consistent with noise, versus identifying it as an SPSD candidate is

$$u\left(N_s, f_{\rm FP}\right) = \left(\varepsilon \mid P(\max \geq \varepsilon \mid N_s \text{ samples}) = f_{\rm FP}\right).$$

The false positive rate sets the threshold because it is the probability level at which we are willing to accept an upward fluctuating noise as an SPSD candidate. for the default, $f_{FP} = 0.005$, and $N_s = 4634$, the numerical solution obtained by the algorithm is $u = 4.74\sigma$. The median, $f_{FP} = 0.5$, of the maxima distribution in a window,$N_s = 193$, is also used in the criteria, giving $u = 2.69\sigma$.

Since Kepler is all about transits, we take steps to ensure we do not misidentify any transits as SPSDs. Assuming the maximum exceeds the 0.005 false positive probability threshold, we address transits in two ways, both of which involve identifying the minimum detection statistic in a $N_{\rm LongWindow}$-cadence-long window, centered at the cadence when the maximum occurred:

- if the probability of accidental occurrence of the observed sum of the maximum and minimum is smaller than our selected false positive rate, then the SPSD is vetoed, or
- if the minimum is statistically equal and opposite in sign to the maximum, then the SPSD is vetoed.

These criteria are discussed in §2.2. The first of these criteria requires an additional threshold calculation involving joint probabilities. We note, as above that the cumulative distribution function for a maximum in N_1 samples is

$$P(\max(x_{\rm obs}) \geq x \mid N_1 \text{ samples}) = 1 - P(x_{\rm obs} \mid x_{\rm obs} \leq x)^{N_1}.$$

Likewise, because the PDF is symmetric, the cumulative distribution function for a minimum in N_2 samples is

$$P(\min(x_{\rm obs}) \leq -y \mid N_2 \text{ samples}) = 1 - P(x_{\rm obs} \mid x_{\rm obs} \leq y)^{N_2}.$$

The joint PDF for given maximum, x and given minimum, $-y$ (y positive), is the 2-dimensional product of $dP(max\ldots)/dx \times dP(min.\ldots)/dy$. Given that we've already constrained the maximum, we can contract this product to a one-dimensional distribution on the sum $\Sigma = x + (-y)$ by setting $(-y) = -x + \Sigma$, and

then integrating over all x. So the probability of the sum of maximum and minimum, being $\geq \Sigma$ for a maximum x in N_1 samples and minimum $-x+\Sigma$ in N_2 samples is

$$P(\max(x_{\rm obs1}) + \min(x_{\rm obs2}) \geq \Sigma_t h \mid N_1(N_2) \text{ samples in obs1(obs2)}) = \int_{-\infty}^{\infty} \frac{dP(\max(x_{\rm obs}) \geq x | N_1 \text{ samples})}{dx} \frac{dP(\min(x_{\rm obs}) \leq -x+\Sigma | N_2 \text{ samples})}{dx} dx \ .$$

So, as above, the threshold on the sum Σ (or difference $x-y$) for a give false positive rate is

$$u_\Delta\left(N_1, N_2, f_{\rm FP}\right) = \left(\Sigma \mid P(\max(x_{\rm obs1}) + \min(x_{\rm obs2}) \geq \Sigma \mid N_1(N_2) \text{ samples in obs1(obs2)}) = f_{\rm FP}\right).$$

For defaults $\{N2 = N_{\rm LongWindow} = 193,\ f_{FP} = 0.005\}$, and $N_s = 4634$ we obtain numerically, $u_\Delta = 2.28$. If the sum Σ is less than this, the SPSD is vetoed as a candidate.

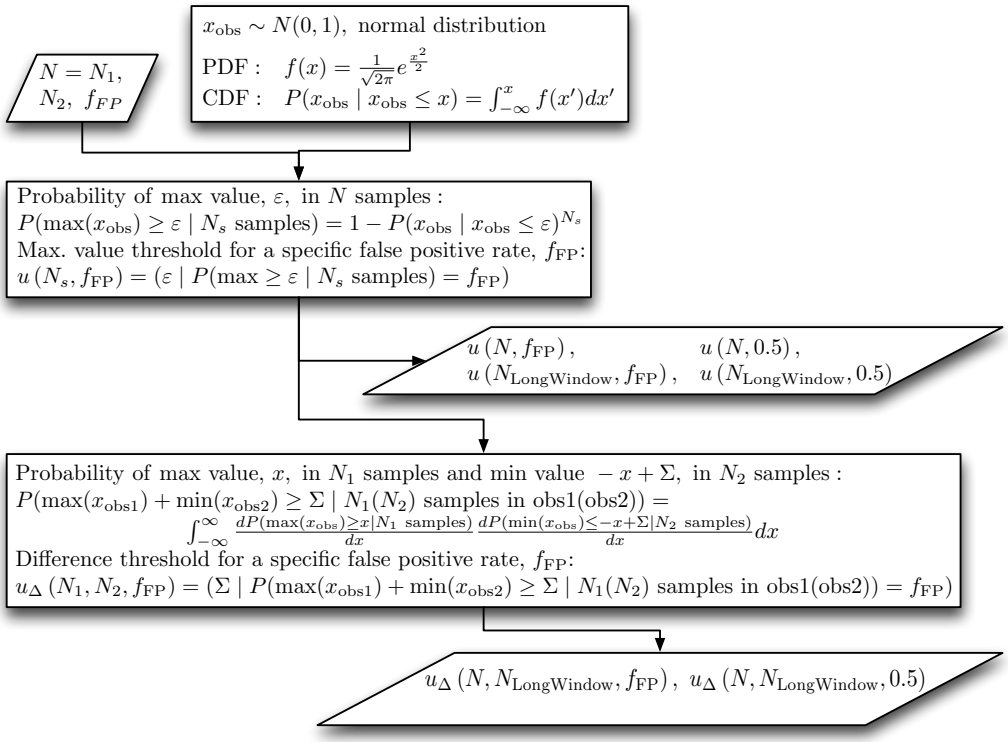


FIGURE 37. **Extreme value threshold calculations for candidate identification segments.** The algorithm requires a set of parameters: the time series length, N or N_1 cadences; a window length $N_2(= N_{\rm LongWindow})$ cadences; the accepted fractional rate of false positive SPSD detections, $f_{\rm FP}$. The algorithm produces 6 extreme value thresholds for 3 sample conditions and 2 false positive rates: $u\left(N, f_{\rm FP}\right)$ is the $f_{\rm FP}$ probability extreme (max) value for N-sample series; $u\left(N, 0.5\right)$ is the 50% probability extreme (max) value for N-sample series; $u\left(N_{\rm LongWindow}, f_{\rm FP}\right)$ is the $f_{\rm FP}$ probability extreme (max) value for $N_{\rm LongWindow}$-sample series; $u\left(N_{\rm LongWindow}, 0.5\right)$ is the 50% probability extreme (max) value for $N_{\rm LongWindow}$-sample series; $u_\Delta\left(N, N_{\rm LongWindow}, f_{\rm FP}\right)$ is the $f_{\rm FP}$ probability difference in extreme values (max-max or max+min) for one value in an N-sample series and the other in an $N_{\rm LongWindow}$-sample series; and $u_\Delta\left(N, N_{\rm LongWindow}, 0.5\right)$. is the 50% probability difference in extreme values (max-max or max+min) for one value in an N-sample series and the other in an $N_{\rm LongWindow}$-sample series. The detection thresholds are derived from a Gaussian probability distribution. See fig. 13.

Appendix C. Details of Sinusoid Correction Algorithm

In the correction algorithm described in § 2.3, we identify sinusoids using the algorithm segment diagramed in fig. 38. This shares the false positive statistics derived in Appendix B. We find candidates based on a robust standardized distribution of the Fourier amplitude first differences, (e.i. $x_{i+1} - x_i$). This is followed by a confirmation step which reevaluates the noise in a 100 FFT-bin band around each candidate. The result is a list of candidates ordered by frequency.

We model sinusoidal component using the algorithm segment diagramed in fig. 39. The algorithm tries to model a complete waveform at given period rather than just a single sinusoid. Starting with a reference frequency, ν_X, it models harmonics of ν_X up to the maximum candidate frequency and subharmonics down to frequency, 1/(window length). The subharmonics help to cover the cases where the waveform period is longer by some integral number than $1/\nu_X$. The algorithm performs a nonlinear fit to a precision specified by the range and number of frequencies in a scan of the base frequency to find the best-fit frequency, ν_Y which minimizes the robust standard deviation of residuals.

Fig. 19 indicates how these two segments are applied within the overall algorithm.

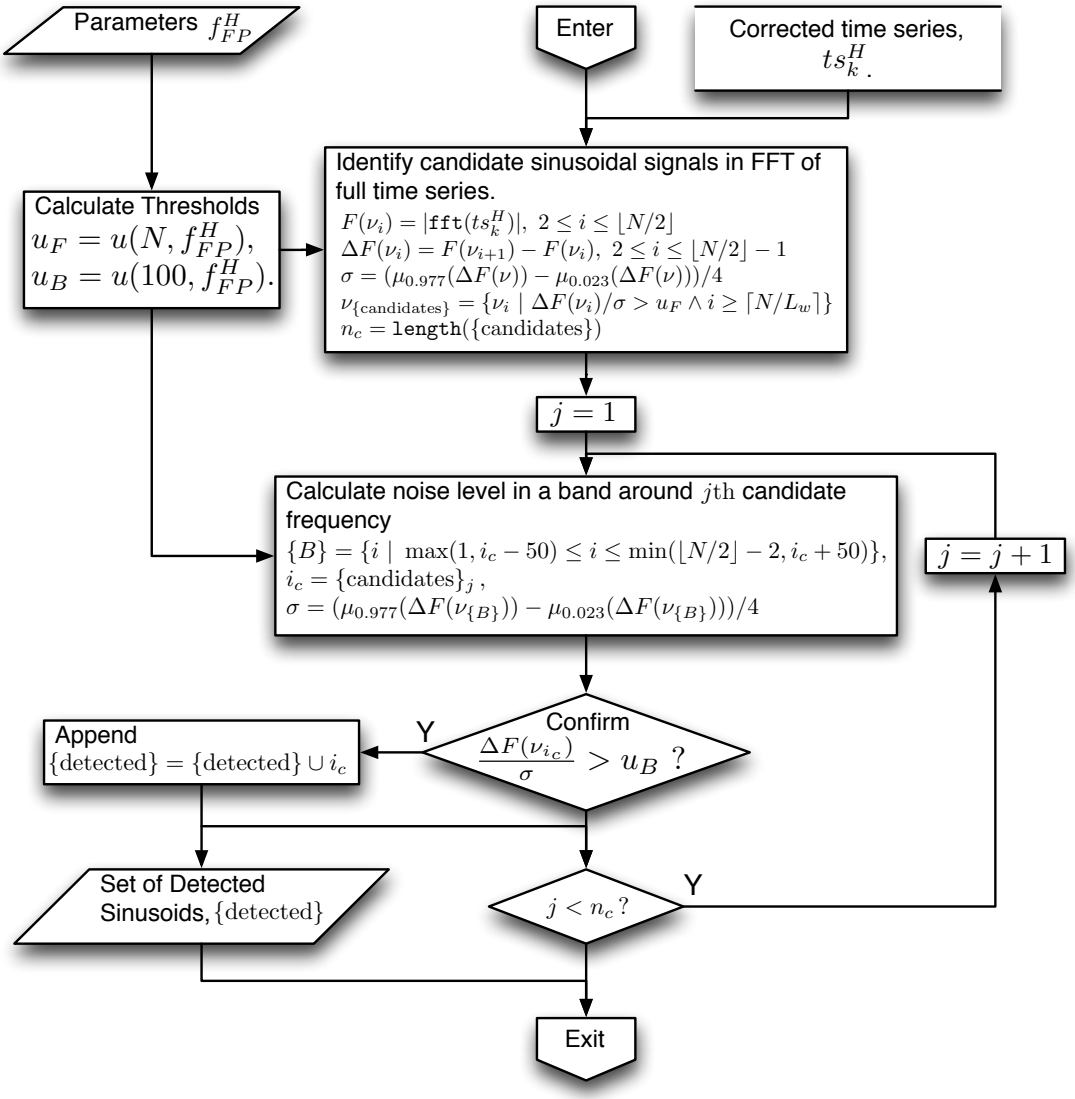


FIGURE 38. **Sinusoid identification for sinusoid removal segment.** This segment requires an input parameter, f_{FP}^H, which specifies the acceptable rate of false positive sinusoid detections, and a partially corrected flux time series, ts_k^H. It produces a set of confirmed sinusoid signals, $\{\text{detected}\}$. The process first identifies candidate frequencies based on the standardized first difference of FFT amplitudes exceeding a threshold for the full amplitude spectrum, u_F. It then confirms or rejects the candidates based on re-standardizing in a ± 50 FFT-bin band surrounding each candidate. Confirmation requires that the re-standardized amplitude at a candidate frequency exceed a threshold for the band, u_B. Both u_F and u_B are derived from f_{FP}^H based on the equation for $u\left(N_s, f_{\text{FP}}\right)$ in Appendix B. $\mu_X(\text{data})$ represents the X quantile of data. The list of detected frequencies are limited in precision by the FFT bin-width and are refined in later steps of the sinusoid removal algorithm. See fig. 19.

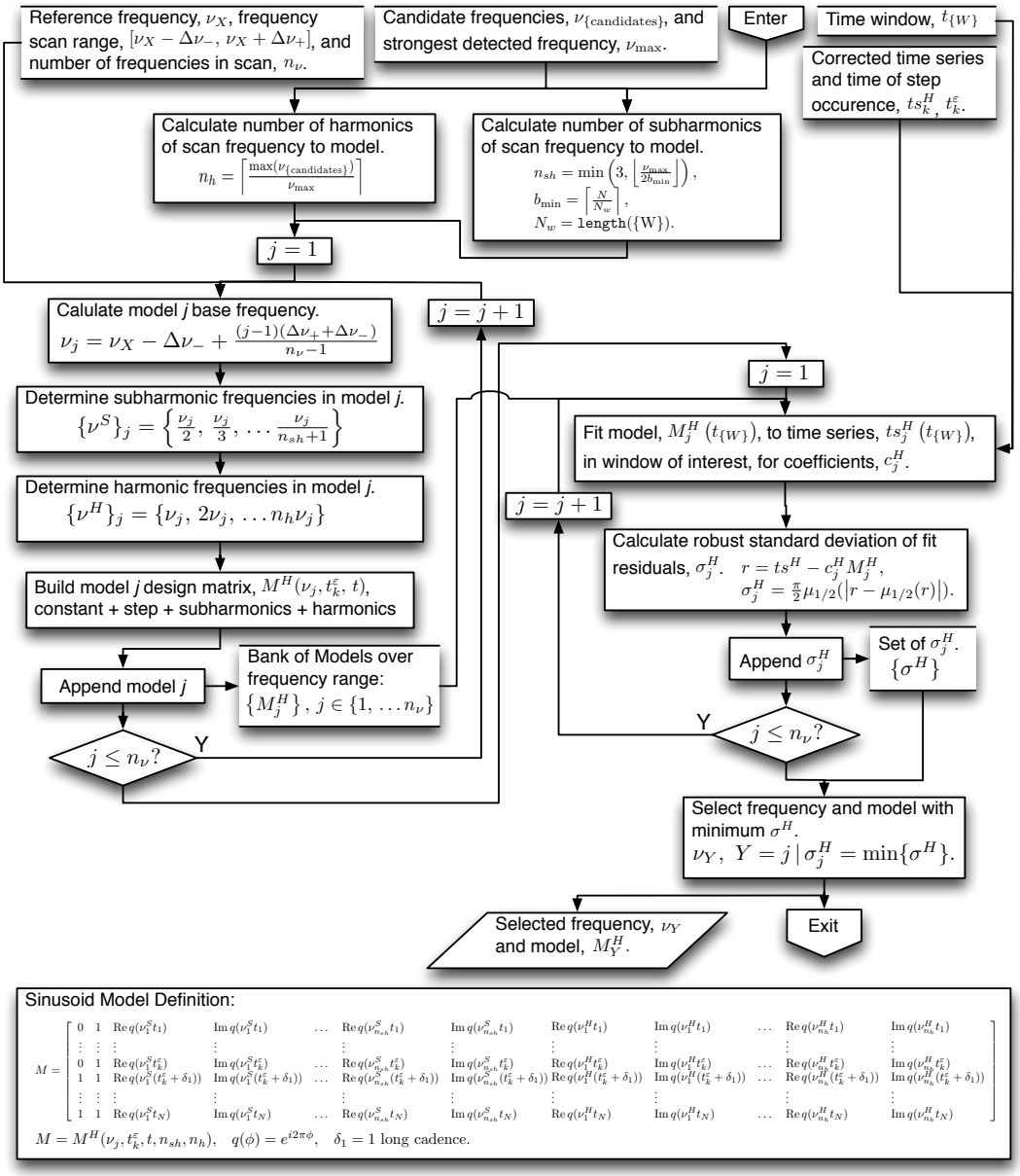


FIGURE 39. **Sinusoid modeling for sinusoid removal segment.** This segment requires a reference frequency, ν_X, a scan range, $[\nu_X - \Delta\nu_-, \nu_X + \Delta\nu_+]$, consisting of n_ν frequencies, the set of candidate frequencies, $\nu_{\{\text{candidates}\}}$, from the FFT segment described in fig. 38 the frequency in $\nu_{\{\text{detected}\}}$ with the largest detection above threshold, $\nu_{\max}$, the fitted time window, $t_{\{W\}}$, and a partially corrected flux time series, ts_k^H with SPSD occurence time, t_k^ε. It produces a selected frequency, ν_Y, which gives the best fit among the frequencies scanned, and the model, M_Y^H, associated with ν_Y. The process first constructs a bank of models, each of which includes a range of both subharmonics and harmonics of the base frequency as well as a step. It then fits the models to the data in a window around the SPSD, calculating a robust standard deviation of the fit residuals, σ_j^H for the jth base frequency in the scan range. Finally, it selects ν_Y as the frequency with the smallest σ_j^H. This algorithm segment is applied multiple times for each of up to 3 separate frequencies. See fig. 19.